T H E S E

DE DOCTORAT D'ETAT ES SCIENCES MATHEMATIQUES

présentée à

L'UNIVERSITE DE PARIS VI

par

Claude ROCHE

pour obtenir le grade

de

DOCTEUR ES SCIENCES

INFORMATION UTILE EN RECONNAISSANCE DES FORMES ET EN COMPRESSION DE DONNEES APPLICATION A LA GENERATION AUTOMATIQUE DE SYSTEMES DE RECONNAISSANCE OPTIQUE ET ACOUSTIQUE

Soutenue le 9 Décembre 1972 devant la Commission d'examen

Jury :	P. LELONG	Président
	L. MALAVARD	Examinateur
	J. C. SIMON	Rapporteur
	C. F. PICARD	Invités
	W. MERCOUROFF	

T H E S E

DE DOCTORAT D'ETAT ES SCIENCES MATHEMATIQUES

présentée à

L'UNIVERSITE DE PARIS VI

par

Claude ROCHE

pour obtenir le grade

de

DOCTEUR ES SCIENCES

INFORMATION UTILE EN RECONNAISSANCE DES FORMES ET EN COMPRESSION DE DONNEES APPLICATION A LA GENERATION AUTOMATIQUE DE SYSTEMES DE RECONNAISSANCE OPTIQUE ET ACOUSTIQUE

Soutenue le 9 Décembre 1972 devant la Commission d'examen

Jury :	P. LELONG	Président
	L. MALAVARD	Examinateur
	J. C. SIMON	Rapporteur
	C. F. PICARD	Invités
	W. MERCOUROFF	

UNIV.
LEIDEN
BIBL.

Je remercie vivement Monsieur le Professeur P. LELONG de l'honneur qu'il m'a fait en présidant le jury de cette thèse. Je remercie aussi Messieurs les Professeurs L. MALAVARD et W. MERCOUROFF ainsi que Monsieur C. F. PICARD, Directeur de Recherches, de l'intérêt qu'ils ont manifesté pour mon travail et d'avoir bien voulu constituer le jury.

Je dois des remerciements particuliers à Monsieur le Professeur J. C. SIMON grâce à qui ce travail a trouvé une existence. Ses conseils, ses idées, son dynamisme m'ont été d'une aide incomparable.

Je remercie Monsieur l'Ingénieur Général DERAMOND et la D. T. A. T. de m'avoir permis d'effectuer librement ces travaux de recherche.

Je tiens également à remercier Messieurs SABAH, DAVANCENS, HUET, SECHET, POUSSIN, RAABE, ROSSIGNOL, HEUDE, LAUZANNE, MLOUKA et Madame HUBERT dans leur aide à la réalisation des programmes.

Madame RIVIERE a eu la lourde tache de la dactylographie et Madame HANNEQUIN et son équipe celle du tirage et de la reliure. Je les remercie de leur gentillesse et de l'excellent travail qu'elles ont fait.

"Mais comment peut-on passer par une porte"

Norbert WIENER - 1948
(en remarque à la fin d'une conférence
sur la relativité généralisée)

INFORMATION UTILE EN RECONNAISSANCE DES FORMES ET EN COMPRESSION DE DONNEES

APPLICATION A LA GENERATION AUTOMATIQUE DE SYSTEMES DE RECONNAISSANCE OPTIQUE ET ACOUSTIQUE

INTRODUCTION

CHAPITRE I : Notion d'opérateur en Reconnaissance des Formes

I.1. Les opérateurs

I.1.1. Les capteurs
I.1.2. Les ordinateurs
I.1.3. Nomenclature
I.1.4. La reconnaissance des formes et la notion d'opérateur

I.2. Notion de niveaux différents d'opérateurs - Exemples

I.2.1. Niveaux différents de formes : la reconnaissance de lettres et de mots
I.2.2. L'opérateur considéré comme un assemblage d'opérations : les filtres linéaires
I.2.3. Niveaux différents d'opérateurs

I.3. Opérateurs syntaxiques

I.4. Formalisation des opérateurs et structures d'opérateurs

I.4.1. Définition formelle des opérateurs
I.4.2. Définition formelle des structures d'opérateurs
I.4.3. Notation formalisée des structures de reconnaissance

CHAPITRE II : Les opérateurs de commande

II.1. Notion d'opérateur de commande

II.1.1. Notion de paramètre d'un opérateur

II.1.2. Les opérateurs de commande

II.2. Exemples d'opérateurs de commande

II.2.1. L'opérateur distance euclidienne

II.2.2. Autres exemples

II.3. Définition formelle des opérateurs de commande

II.3.1. Introduction

II.3.2. Notions de variables matérielles, d'opérateurs

II.3.3. Les opérateurs de commande

CHAPITRE III : Information utile apportée par un opérateur

III.1. L'opérateur idéal

III.2. Probabilités et théorème de Bayès

III.3. Néguentropie et information mutuelle

III.4. Mesure de la qualité des caractéristiques en Reconnaissance des formes

III.5. Mise en série et en parallèle d'opérateurs

III.5.1. Mise en série d'opérateurs

III.5.2. Mise en parallèle d'opérateurs

III.5.3. Exemple de l'addition

III.6. Relation entre la probabilité d'erreur et l'information utile

III.7. La mesure des probabilités et des fonctions néguentropie et information mutuelle

III.8. Exemples de mesure de l'information utile

III.8.1. Capteur optique
III.8.2. Capteur acoustique (corrélation intercanaux)
III.8.3. Capteur acoustique (reconnaissance de phonèmes)
III.8.4. Capteur acoustique (segmentation)

III.9. Validité de l'utilisation de l'information utile

CHAPITRE IV : Utilisation de l'Information Mutuelle dans des algorithmes séquentiels de Reconnaissance de formes

IV.1. Les Pseudoquestionnaires

IV.2. Reconnaissance de mots

IV.3. Reconnaissance de lettres

IV.4. Reconnaissance de la parole

IV.5. Causes d'erreur de la méthode

CHAPITRE V : Méthode de génération automatique d'opérateurs par génération aléatoire et sélection progressive

V.1. Génération automatique d'opérateurs de prétraitement

V.3. Cas particulier des opérateurs à seuil

V.4. Conclusion - Applications pratiques

CHAPITRE VI : Méthode de génération automatique d'opérateurs par leur table

VI.1. Principes généraux

VI.2. Génération de la table de l'opérateur

VI.2.1. Méthode de regroupement employée

VI.2.2. Etude de regroupements utilisant d'autres critères que l'information mutuelle

VI.2.2.1. Critère $H[p[A_1 \circ A_2]$ maximum

VI.2.2.2. Critère $H[p[A_1 \circ A_2]$ minimum

VI.2.2.3. Remarques sur ces deux critères

VI.2.2.4. Critère $I[p[A_1 \circ A_2] ; A_3]$ minimum

VI.2.2.5. Critère $I[p[A_1 \circ A_2] ; A_3]$ maximum

VI.2.2.6. Critère $I[p[A_1 \circ A_2] ; \Omega] - I[p[A_1 \circ A_2] ; A_3]$ maximum

VI.3. Génération de la structure

VI.3.1. Principe et description d'un programme général de création de structures de reconnaissance

VI.3.2. Introduction des coûts associés aux formes

VI.3.3. L'étude de l'ergodicité

VI.3.4. Programmation de l'algorithme d'apprentissage et ses performances

VI.4. Problème de la reconnaissance de phonèmes (données vocales)

VI.5. Problème de la segmentation de la parole (données vocales)

VI.6. Reconnaissance de mots à partir de phonèmes : prétraitement (données fournies par un système de reconnaissance de phonèmes)

VI.6.1. Introduction du problème
VI.6.2. Les opérateurs diphonèmes simples
VI.6.3. Les opérateurs diphonèmes améliorés
VI.6.4. Factorisation des opérateurs de position
VI.6.5. L'étage de reconnaissance de mots

VI.7. Reconnaissance des forêts sur photos aériennes

VI.8. Apprentissage d'une note de ressemblance

VI.8.1. Introduction
VI.8.2. Cadre du problème
VI.8.3. Résolution du problème
VI.8.4. Conclusions

VI.9. Remarque sur la correspondance entre les formes et les réponses de la structure engendrée

CONCLUSION ET PROSPECTIVE

REFERENCES

ANNEXES

INTRODUCTION

La "reconnaissance des formes" a été au départ un ensemble de techniques principalement informatiques (mais aussi optiques) qui doit permettre de classer des images telles que lettres imprimées, chiffres... Puis le sens du mot forme s'est étendu, et l'on a appelé "reconnaissance de formes" le fait de classer automatiquement des sons, des données provenant d'appareil de physique etc... Par extension, on peut considérer actuellement comme problèmes de "reconnaissance de formes", les problèmes de partage de données en classe, dans le but de satisfaire un certain critère, en général celui d'approcher le partage subjectif de ces données par un être humain.

Pour résoudre un problème particulier, les informaticiens ont l'habitude de l'analyser par un effort de réflexion, d'en faire l'organigramme puis d'écrire le programme correspondant qui donnera la solution grâce à un ordinateur... Ici, l'analyse se révèle en général infructueuse. Du moins conduit-elle souvent à une recherche approfondie et donc longue, des phénomènes mis en oeuvre. De plus, même dans les cas où ces phénomènes sont décrits, des facteurs aléatoires interviennent et il n'existe souvent pas d'algorithme résolvant le problème de manière satisfaisante.

Des méthodes générales de reconnaissance des formes seraient donc souvent très utiles. Malheureusement, au stade actuel des recherches, ce ne sont encore que des procédés divers utilisant d'autres théories : probabilités, grammaires. De nombreux scientifiques ont pourtant cherché à élaborer une théorie de la reconnaissance des formes :

Certains ont proposé des méthodes de classification par hypersurfaces : une donnée est représentable par un point d'un espace à n dimensions. Partager des données revient à trouver des hypersurfaces séparant les "nuages" de points qu'on veut séparer [7][67]. Cette théorie qui a donné lieu aux études sur le Perceptron s'est apparemment avérée insuffisante pour la résolution de certains problèmes [50].

D'autres ont utilisé la théorie des probabilités, par des classificateurs probabilistes utilisant en général la formule de Bayes et les notions de coût et de probabilité d'erreur. TZYPKIN [89] a montré les liens entre les deux points de vue.

De nouvelles méthodes typiquement informatiques, dites de "parcours d'arbre" apparentées aux recherches en "Intelligence Artificielle" ont conduit SLAGLE [86] et d'autres à de nouveaux classificateurs.

Au lieu de classer directement les données provenant de l'appareil de mesure, on a voulu introduire la notion de prétraitement et d'extraction de "primitives", c'est-à-dire transformer les données de départ en données classifiables par l'une des méthodes précédentes. En général, un modèle des formes est construit, les paramètres du modèle sont déterminés dans chaque cas, par exemple par identification, la valeur de ces paramètres fournit les données à classifier.

Vers 1964, à la suite du courant provoqué par les idées de CHOMSKY sur les grammaires formelles, on a voulu [37],[52] appliquer cette nouvelle théorie à la reconnaissance des formes. FU [87] a relié ces idées aux théories probabilistes et a proposé la notion de grammaire stochastique.

Le problème de la classification a priori (sans "professeur") s'est introduit en même temps, par les travaux de WATANABE [90], COOPER, etc sans nécessairement beaucoup de liens avec les autres méthodes ...

Apparemment, la "Reconnaissance des Formes" se cherche encore en tant que théorie unifiée.

Plusieurs remarques peuvent être faites à propos de ces recherches :

1. La plupart des publications dans ce domaine présentent des idées, mettent en oeuvre des théories, n'ayant que des liens très lointains avec les applications réelles de la reconnaissance des formes : les problèmes résolus par ces nouvelles idées, s'ils ne sont pas triviaux, ne sont pas en général pratiques.

2. Toutes les méthodes de reconnaissance nécessitent soit l'utilisation d'un ordinateur, soit la création d'un système spécialisé mis au point en général par simulation sur ordinateur. La plupart des publications ne semblent pas prêter beaucoup d'attention au lien fondamental qui existe entre tout problème de reconnaissance et le support matériel de la reconnaissance ; et ceci non point tant du point de vue de la résolution pratique du problème, mais des bases théoriques même des méthodes envisagées.

3. A part UHR et VOSSLER et les études de FELDMANN [94] sur les "inférences de grammaires formelles", tous les mécanismes d'apprentissage optimisent des coefficients : cet ajustement peut-il être considéré comme un apprentissage ? Nous pensons qu'actuellement l'apprentissage des systèmes de reconnaissance se fait dans le cerveau du chercheur : l'ordinateur ne sert que de support au système trouvé, en déterminant quelquefois ses paramètres. Les grandes possibilités des calculateurs électroniques sont en fait utilisées de manière très faible.

Pourquoi ne pas confier un rôle d'apprentissage à l'ordinateur lui-même ? Il est capable d'essayer dans un temps beaucoup plus rapide que l'homme de nombreuses possibilités pour aboutir à un bon système de reconnaissance.

Les études que nous présentons ici sont un effort dans les directions appelées par ces trois remarques :

Nous considérons comme essentiel le fait qu'un système de reconnaissance doit avoir un support matériel dont le type le plus général est l'ordinateur.

Nous proposons des méthodes d'apprentissage différentes des méthodes usuelles, et fortement inspirées par les principes de fonctionnement des machines automatiques.

Enfin, nous avons appliqué les méthodes proposées à des exemples pratiques et nous en avons montré les performances sur ces exemples.

Le premier chapitre présente la notion d'opérateur de reconnaissance et de structure d'opérateurs de reconnaissance. Sur des exemples que nous avions précédemment étudiés, nous montrons comment la plupart des mécanismes de reconnaissance font appel à ces structures. Nous proposons une formalisation de ces notions nouvelles. Nous montrons aussi comment tout problème de reconnaissance fait appel à ces notions de manière fondamentale, de même que le principe de fonctionnement de toute machine automatique, par exemple un ordinateur.

Le deuxième chapitre présente la notion nouvelle d'opérateur de commande très liée à la structure matérielle des machines automatiques, et à la recherche en mémoire de données dans ces machines. Ces opérateurs de commande sont la raison principale de la souplesse des ordinateurs par rapport aux machines automatiques simples.

Dans le troisième chapitre, nous introduisons la fonction d'évaluation de ces opérateurs qui sera utilisée systématiquement : l'"information utile". Cette fonction introduite par LEWIS [41] est justifiée ici

1. par la théorie des questionnaires,
2. par le fait que les mesures faites dans des exemples pratiques corresdent aux notions intuitives d'"information utile",
3. par l'existence de trois relations compatibles avec ce qu'on attend d'une mesure de l'"information utile". Nous démontrons ces trois relations :

A notre connaissance, aucun auteur n'a signalé l'existence de la troisième de ces relations.

Le quatrième chapitre présente une première utilisation de cette fonction "information utile" dans des algorithmes séquentiels de reconnaissance. La reconnaissance de lettres manuscrites, de mots et de phonèmes (reconnaissance vocale) a été traitée.

Une première méthode d'apprentissage est proposée dans le cinquième chapitre. Elle utilise une procédure de choix aléatoire d'opérateurs et de sélection successive des meilleures structures construites. La sélection s'effectue suivant les plus grandes valeurs de l'information utile. Cette nouvelle méthode a permis la génération automatique de filtres linéaires, et sur une une autre application, de la fonction distance euclidienne.

La méthode d'apprentissage que nous avons principalement étudiée est décrite dans le sixième chapitre. Elle fait appel à l'information utile à la fois dans la création de chaque opérateur de la structure, et dans la création de la structure elle-même. Plusieurs problèmes pratiques de reconnaissance ont été résolus par cette méthode. Les performances obtenues sont meilleures que celles des méthodes classiques de reconnaissance telles que partage par hypersurfaces, ou discrimateurs Bayesiens. Elles sont même parfois bonnes lorsque les méthodes classiques ne sont pas utilisables. Nous présentons ces applications et les résultats obtenus dans le sixième chapitre.

Ces méthodes originales d'apprentissage automatique semblent très prometteuses à la fois sur le plan des applications pratiques, et sur le plan des idées théoriques mises en oeuvre. Elles semblent pouvoir donner lieu à des développements ultérieurs importants, amenant à traiter du problème de l'apprentissage généralisé appliqué à d'autres problèmes que des problèmes de reconnaissance de formes. Le dernier paragraphe du chapitre VI présente déjà un problème différent de la reconnaissance des formes et résolu par notre même programme d'apprentissage.

CHAPITRE I

NOTION D'OPERATEUR EN RECONNAISSANCE DE FORMES

I.1. LES OPERATEURS

I.1.1. Les capteurs

La Reconnaissance de Formes étudie certains traitement de données. Ces Ces données sont fournies par des appareils de mesure appelés capteurs.

Un tel capteur est le "CALIFE" appartenant au Laboratoire Central de l'Armement. C'est un capteur optique constitué d'un "flying spot" balayant une photographie diapositive 24 x 36. Des plaques électrostatiques permettent de focaliser un faisceau électronique en un point donné de la photographie. La plage éclairée module ce faisceau et après amplification et digitalisation du faisceau, on dispose d'une valeur discrète :

$$A(x, y)$$

Elle représente le résultat d'une mesure de l'intensité de la photographie au point (x, y).

On paramétrise cette mesure par le couple (x, y) dans le but d'utiliser le même appareil de mesure à différents points de la photographie.

Un ensemble fini de telles mesures peut être "caractéristique" d'un objet "dont on a pris la photographie". Le terme "caractéristique" sera défini au chapitre II.

Le vocoder est un exemple de capteur acoustique. Un vocoder assez précis délivre 36 valeurs toutes les 10 ms. Ces valeurs représentent des mesures de l'énergie sortant de filtres acoustiques, passe bande dont les 36 fréquences centrales sont échelonnées le long du spectre audible.

Un autre capteur acoustique est un discrétiseur en ligne à la fréquence de 10 kHz du signal acoustique.

Ces capteurs transforment un champ de pression mécanique en données digitales.

Un capteur optique ou acoustique est un opérateur qui, à un certain champ électromagnétique ou acoustique associe une valeur discrète ou une série de valeurs discrètes.

Les résultats de cet opérateur ou attributs constituent les données introduites en ordinateur en vue d'un traitement.

Différentes études ont été faites sur des capteurs optiques particuliers : celles de G. J. BALM [1], I. SHEINBERG [74], MORI [51]. Une étude générale sur les capteurs existe dans un numéro spécial de Pattern Recognition (Juillet 1971).

Des études théoriques intéressantes ont été faites sur les erreurs faites par des capteurs optiques, lors de la digitalisation de fonctions continues. Se rapporter plus particulièrement à SCHÖNBERG [71], HOUDAS HORVAT [28] et voir [79] [33] [25]. Ces études permettent de déterminer la précision à atteindre sur les dimensions du maillage de digitalisation et sur la digitalisation en chaque point, pour reconstituer la fonction continue correspondante avec une erreur inférieure à une valeur donnée.

I.1.2. Les ordinateurs

Nous disposons d'une autre classe concrète d'opérateurs, les ordinateurs. Chacune des instructions machine de l'ordinateur est un opérateur qui à un certain état de l'ordinateur fait correspondre un autre état.

Ces opérateurs sont munis eux-aussi d'un ou plusieurs paramètres qui représentent les registres ou les cellules de mémoire sur lesquels ils vont travailler. L'existence de tels paramètres permet d'utiliser le même opérateur cablé pour tous les registres et les cellules de mémoire de l'ordinateur à condition de l'utiliser à des moments différents, et en faisant varier ces paramètres.

Un programme indique à l'ordinateur sous forme de données décodables la manière dont les opérateurs vont travailler : l'indication de l'ordre d'utilisation et des paramètres affectés à chacun d'eux.

Voir le chapitre I de [10].

Remarques :

- Les opérateurs considérés ici sont simples et les correspondances qu'ils établissent sont celles du calcul arithmétique élémentaire :

$$(x, y) \xrightarrow{\text{addition}} (x + y)$$

ou du calcul booléen, ou ce sont des "décalages binaires".

- Par contre, la manière dont ces opérateurs sont enchainés est en général très complexe.

Un programme d'ordinateur ou un sous-programme peut lui-même être considéré comme un opérateur qui à partir des résultats des opérateurs élémentaires donne un résultat qui est l'état des registres et de la mémoire de l'ordinateur à l'arrêt du programme.

I.1.3. Nomenclature

Le résultat d'un opérateur est appelé attribut des données introduites relativement à cet opérateur, cf. BENZECRI [2] qui utilise la notion d'attribut en reconnaissance des formes et traitement de données mais ne considère pas l'opérateur qui le produit.

Equivalence du mot opérateur avec : processeur, algorithme, automate (sens plus restreint), règle de récriture (sens plus restreint), extracteur de paramètres, appareil de mesure, fonction, application.

Equivalence du mot attribut avec : nom d'une forme, nom de primitive, mesure.

Transformée de Fourier, image filtrée sont des attributs relatifs à des opérateurs particuliers appelés FFT et filtre.

I.1.4. La reconnaissance des formes et la notion d'opérateur

La définition usuelle du problème de la reconnaissance des formes est de trouver une méthode automatique qui partage un certain ensemble de données

$$X = \{d_i\}$$

en sous-ensembles D_i ayant chacun un nom $N(D_i)$; le nom correspond à celui que donnerait l'homme à l'une quelconque des données éléments de D_i.

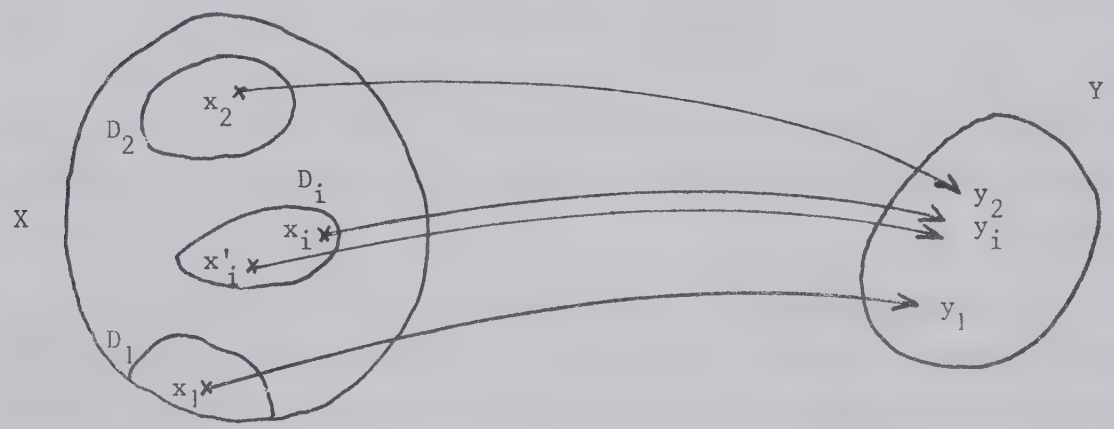


Figure 1

La figure ci-dessus montre d'un côté l'ensemble des données considérées partagé en sous-ensembles (D_1 D_2, ..., D_i, ...) ; de l'autre côté est représenté l'ensemble des noms $\{y_i = N(D_i)\}$.

Si on veut automatiser ce partage en sous-ensembles, il faut automatiser le calcul de la fonction $y_i = N(x_i)$

$$\{N(x_i) = N(D_i), \quad \forall x_i \in D_i\}$$

Autrement dit, créer un opérateur qui à la donnée x_i associe le résultat $y_i = N(x_i)$.

Réciproquement, étant donné un opérateur effectuant une application N d'un ensemble X dans un ensemble Y, cette application induit une partition de l'ensemble X en sous-ensembles d'éléments x_i tels que pour chaque sous-ensemble D_i, il existe un y et un seul tel que $(y = N(x_i) \mid \forall x_i \in D_i)$.

D'autre part, dire que x_i et x'_i sont dans une même classe sous-ensemble de X, est dire qu'il existe une relation d'équivalence entre x_i et x'_i.

Une relation d'équivalence R est telle, rappelons le, que :

- $R\{x, x\}$
- $R\{x, y\} \Rightarrow R\{y, x\}$
- $[R\{x, y\}, R\{y, z\}] \Rightarrow R\{x, z\}$

(voir GODEMENT [22] et SIMON [77]).

On démontre le théorème suivant :

Soit R une relation d'équivalence sur un ensemble E.

Il existe un ensemble E' et une application $f : E \to E'$ tels que les relations :

$$R\{x, y\}$$

$$\text{et} \quad f(x) = f(y)$$

soient équivalentes.

Ce théorème de mathématiques est en parfait accord avec l'utilisation de la notion d'opérateur en reconnaissance de formes.

Remarque :

On doit s'assurer de ne pas faire de pétition de principe au sujet de la définition du signe =. Ce signe implique une relation d'équivalence mais plus restreinte qui échappe d'ailleurs souvent à une définition mathématique. GODEMENT écrit à ce sujet :

"La relation d'égalité a = b signifie quand elle est vraie, que les objets concrets représentés par a et b sont "identiques", notion dont nous ne chercherons pas à approfondir le sens ; l'essentiel, pour le mathématicien, n'est pas d'avoir compris, ou cru comprendre, la "signification profonde" du signe = ; il est de savoir l'utiliser..."

Aussi, voulons nous éviter toute contestation en n'utilisant jamais le signe égal. Nous emploirons dans tout ce qui va suivre seulement des opérateurs. Ils seront définis dans le chapitre suivant soit par leur action quelle

que soit leur entrée (table de l'opérateur), soit par leur structure en utilisant une représentation linguistique.

Mme M. Pavel formalise dans [54] les opérateurs de reconnaissance des formes en en faisant des éléments d'ensembles algébriques. Elle laisse d'ailleurs aux autres le soin de trouver les opérateurs intéressants dans les cas pratiques à résoudre :

"(Un problème que l'on peut se poser) est de réaliser un algorithme de reconnaissance de structures définies ; c'est le plus immédiatement concret, mais aussi le moins général, à chaque type de structures particulières correspondant un algorithme particulier. Nous n'envisagerons pas ce genre de problèmes dans ce cours et renvoyons à ce sujet sur la littérature existante".

La notion d'opérateur intervient de la même manière en intelligence artificielle : H. A. SIMON [76].

Dans ce cas, il s'agit pour prouver un théorème, de trouver une suite d'opérateurs, travaillant en cascade les uns sur les résultats des autres, et reliant les "hypothèses de départ" (données) au but à atteindre : le "théorème à démontrer". L'intelligence artificielle étudie les manières de créer ces chaînes d'opérateurs de manière générale. Elle utilise des fonctions d'évaluations (SAMUEL [69], DELHAYE [14]) mesurant "une distance entre un résultat d'opérateur et le but à atteindre". Nous verrons dans le chapitre IV, une démarche analogue à celle-ci,utilisée dans le but de créer des opérateurs de reconnaissance de forme : la fonction d'évaluation est alors l'information mutuelle.

Le but que nous avons voulu poursuivre, complémentaire des buts de Mme PAVEL, est de trouver des méthodes et des idées générales en vue de créer dans chaque problème particulier, un opérateur adapté de reconnaissance.

I.2. NOTION DE NIVEAUX DIFFERENTS D'OPERATEURS - EXEMPLES

I.2.1. Niveaux différents de formes : la reconnaissance de lettres et de mots

Nous nous sommes posé le problème de la lecture automatique de mots écrits en majuscule baton. Nous avons supposé que le capteur donnait de chaque lettre une représentation telle que sur le premier tableau de la figure 2 : une matrice de 0 et de 1, dont les 1 "dessinaient" la lettre.

Cette étude a été décrite précisément dans [60],[66] et [13].

Plusieurs opérateurs ont été construits :

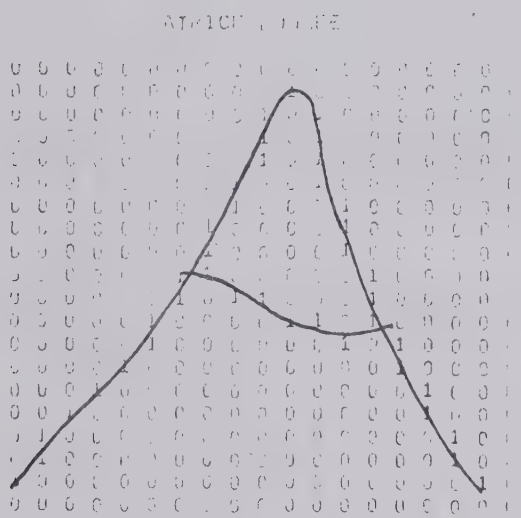
• Le premier opérateur est constitué d'un programme "d'extraction de primitives élémentaires". Il fait passer de cette représentation, à la représentation indiquée par le deuxième tableau de la figure 2. Les points notés 1 dans la première représentation reçoivent un certain numéro ; ceux qui ont le même numéro sont approximativement alignés.

Chacun de ces numéros symbolise une primitive (forme élémentaire), qu'on peut appeler ici "segments rectilignes n° i "du premier type". Ces segments rectilignes du premier type sont tels que leurs relations avec les autres sont des relations de contact par leurs extrémités. Ces relations peuvent être paramétrées par l'angle des deux primitives.

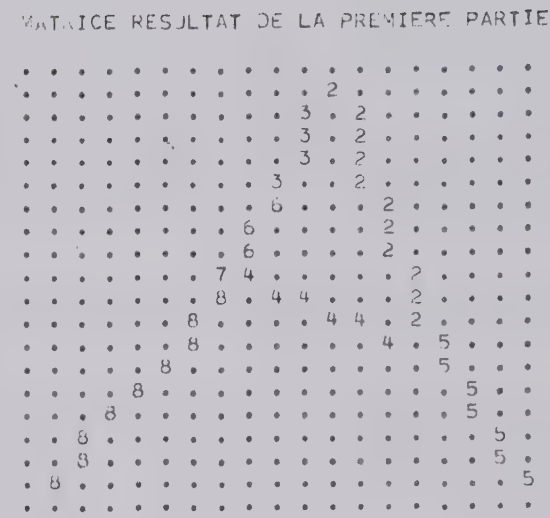
Les lettres sont - intuitivement - caractérisées par ces primitives, leur longueur et leurs relations mutuelles.

• Un deuxième opérateur fait passer de cette dernière représentation à la représentation indiquée par le troisième tableau de la figure 2. Ici, les primitives "extraites" sont les "segments rectilignes du deuxième type", chacun d'eux étant un regroupement de segment du 1er type alignés.

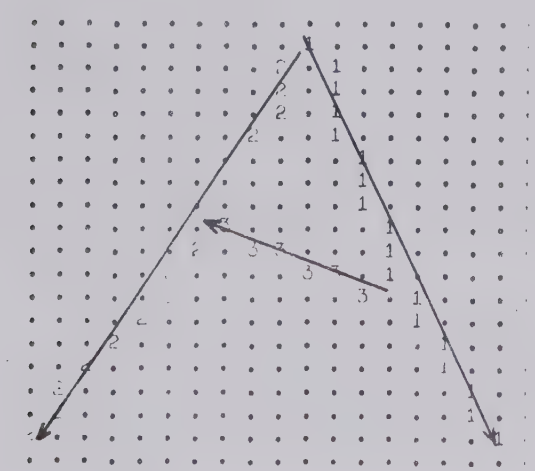
Leur assemblage est décrit par des relations de contact. Celles-ci peuvent être paramétrées par les positions des contacts sur le segment, par l'angle des primitives etc...



MATRICE RESULTAT DE LA PREMIERE PARTIE



MATRICE RESULTAT DE LA SECONDE PARTIE



PRIM1	PRIM2	POS1	POS2	AGL
1	2	0	0	0
1	3	2	0	2
2	3	2	1	1

Figure 2

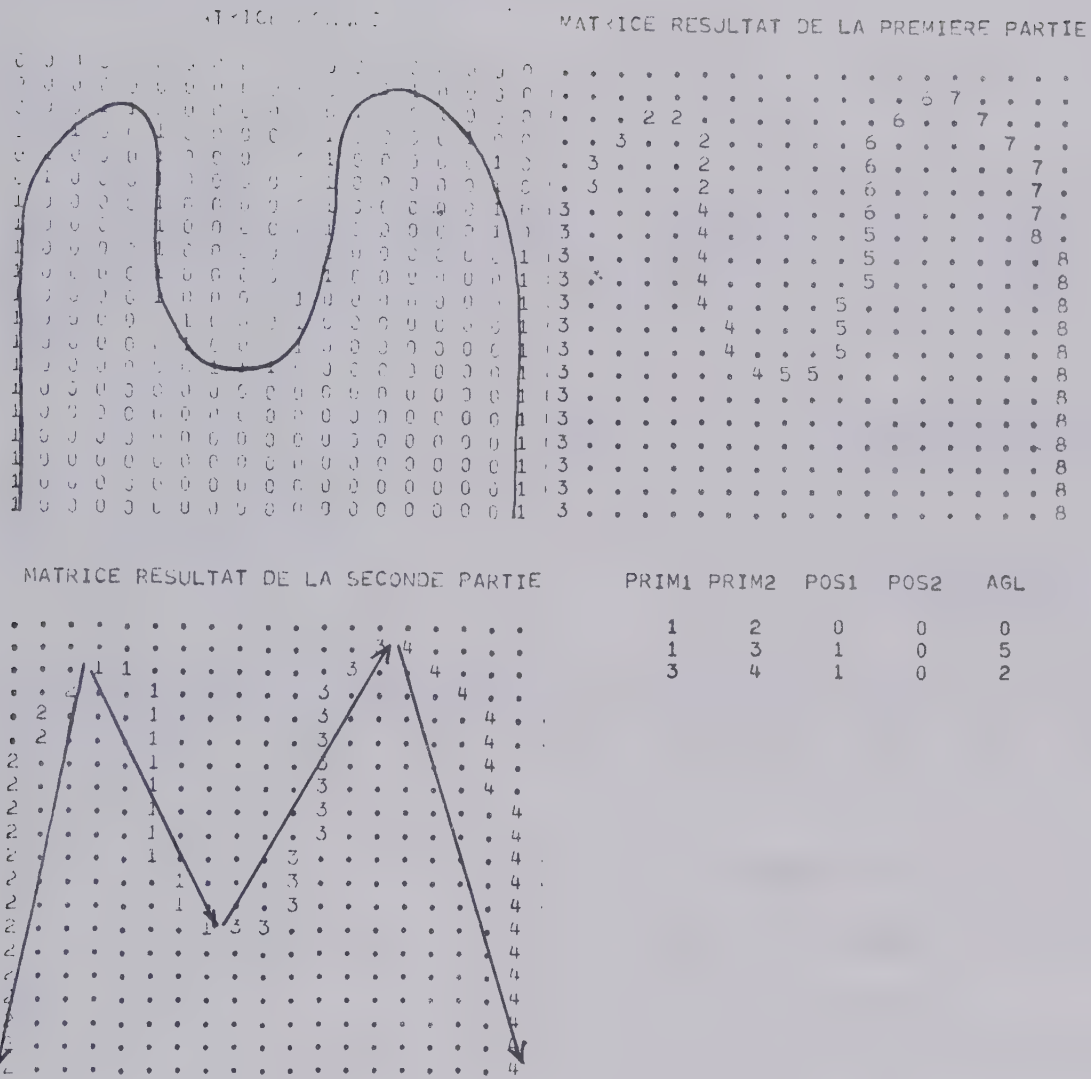

MATRICE RESULTAT DE LA PREMIERE PARTIE
MATRICE RESULTAT DE LA SECONDE PARTIE
PRIM1 PRIM2 POS1 POS2 AGL
1 2 0 0 0
1 3 1 0 5
3 4 1 0 2

Figure 2-bis

• Un troisième opérateur donne de cet assemblage une représentation numérique dont un exemple est donné par le dernier tableau de la figure 2.

Chaque ligne représente une relation entre primitive, la ligne :

1 2 0 0 0

signifie qu'il existe une relation de contact entre les primitives n° 1 et n° 2. Les 0 en 3ème et 4ème colonne signifient qu'elles sont liées par leurs origines, (sur les autres lignes, 1 est le code de l'extrémité, 2 du milieu). Le 0 en 5ème colonne code leur angle en représentation octale.

• Un quatrième opérateur que nous décrirons dans le chapitre IV permet de reconnaître la lettre à partir du codage de cet assemblage.

• Un cinquième opérateur reconnaît les mots grâce à l'assemblage de ces lettres (chapitre IV).

Cette description du système de reconnaissance de mots que nous avons mis au point montre les différents niveaux de forme que nous avons considérés :

- point (extrait par le capteur)
- segment rectiligne du 1er type
- segment rectiligne du 2ème type
- lettre
- mot.

A chacun de ces différents niveaux correspond un opérateur dont les données d'entrée sont les résultats de l'opérateur du niveau précédent.

I.2.2 L'opérateur considéré comme un assemblage d'opérations : les filtres linéaires

1. Soit un opérateur de la forme

$$O_{I,\ J,\ \theta} = \{ \sum_{i,\ j} a_{\theta}(i,\ j) \times P(I + i,\ J + j) > S \}$$

C'est en écriture symbolique la représentation d'un opérateur qui transforme des valeurs numériques par l'intermédiaire des opérations simples codées +, X et >. Ces opérations existent par ailleurs en ordinateur.

C'est un"filtre linéaire".

Nous avons étudié ces opérateurs dans des problèmes pratiques de prétraitement d'images en vue de leur reconnaissance : voir [80] et [81] ainsi que CHECROUN [9].

Les images considérées étaient composées de segments de différentes orientations. Les filtres utilisés avaient pour objet de détecter ces segments, considérés comme formes caractéristiques des images.

Ces opérateurs linéaires avaient deux réponses (oui, non). La réponse (oui) de l'opérateur $O_{I,\ J,\ \theta}$ correspondait à l'existence en (I, J) d'un segment d'orientation θ.

Les coefficients des opérateurs $O_{I,\ J,\ \theta}$ étaient déterminés de telle manière que

- ce qui apparaissait à l'oeil comme appartenant à une même ligne était détecté comme tel par l'ordinateur.
- les lignes de petite taille et le bruit étaient éliminés.

Remarque :

Notons ici que l'intervention de l'observateur humain est indispensable pour juger de la validité d'un opérateur.

Un nombre énorme de publications existe sur les opérateurs linéaires. En reconnaissance des formes, le perceptron en est une application importante. Le perceptron est en effet un réseau de fonctions linéaires et de fonction de seuil. Le réseau est déterminé aléatoirement en plusieurs niveaux et les coefficients sont déterminés par apprentissage. Voir à ce sujet HAY MARTIN et WIGHTAM (1960) [27], ROSENBLATT (1962) [67], BROSSY (1963) [6], LIEGEOIS (1964) [42], CASTAN (1966) [7]. Les résultats des perceptrons ont été satisfaisant dans certains cas, mais les échecs dans des problèmes pratiques ont conduit MINSKY et PAPERT [50] à établir un certain nombre de théorèmes indiquant les limites de validité des systèmes utilisent le principe du perceptron.

D'autres études intéressantes sur les opérateurs linéaires sont dues à TSYPKIN [89] qui relie la détermination des coefficients de l'opérateur à des considérations stochastiques.

AIZERMAN, WIDROW utilisent des opérateurs du second degré avec des méthodes identiques. Ceux-ci utilisent aussi des opérations élémentaires : +, - et X.

Certains procédés d'analyse factorielle conduisent à des opérateurs linéaires, tels les projections sur les plans de plus faible inertie, cf BENZECRI [2].

2. Analysons ce dernier exemple de l'opérateur linéaire :

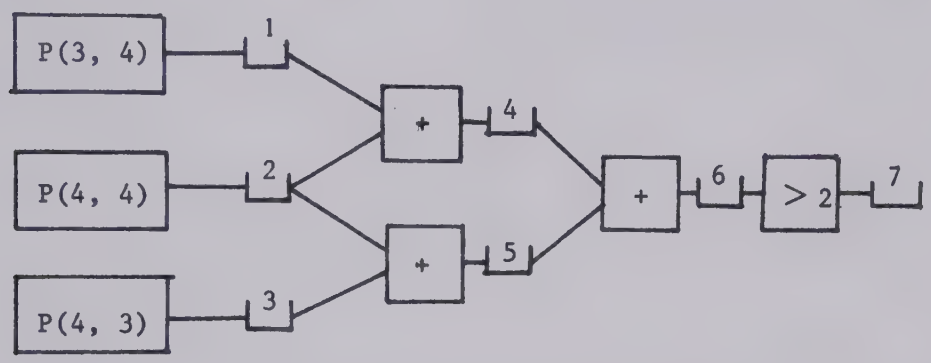
$$\sum_{i,\,j} a_{\theta}(i,\,j)\ P(I + i,\,J + j) > S$$

Que ce soit par un cablage spécialisé, un programme, ou même "à la main" que ce calcul soit effectué, l'opérateur se décompose en opérateurs élémentaires qui travaillent successivement ou en parallèle. Ces opérateurs élémentaires sont des additions et des multiplications ou même seulement des additions. Mais ces opérateurs travaillent sur des données provenant soit d'autres opérateurs identiques soit d'un capteur.

Représentons le travail effectué par le filtre :

$$P(3, 4) + 2P(4, 4) + P(4, 3) > 2$$

on peut représenter chaque opération élémentaire par un rectangle et par un segment les registres dont les états sont modifiés par les opérateurs :



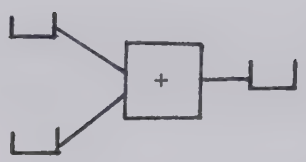
Supposons que :

Les capteurs délivrent des valeurs comprises entre 0 et 3 et qu'ici les registres 1, 2 et 3 contiennent les valeurs 0, 1 et 1. Dans ce cas, le registre 4 contient la valeur 1 de l'ensemble {0, 1, 2, ..., 6}, le registre 5 la valeur 2 de l'ensemble {0, 1, 2, ..., 6}, le registre 6 la valeur 3 de l'ensemble {0, 1, 2, ..., 9}, enfin le registre 7 la valeur oui de l'ensemble {oui, non}.

Chaque opération peut être représentée par sa table ;

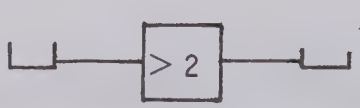
- une opération **binaire** comme l'"addition" par exemple :

	0	1	2	3
0	0	1	2	3
1	1	2	3	4
2	2	3	4	5
3	3	4	5	6



- une opération unaire comme l'"opération seuil" :

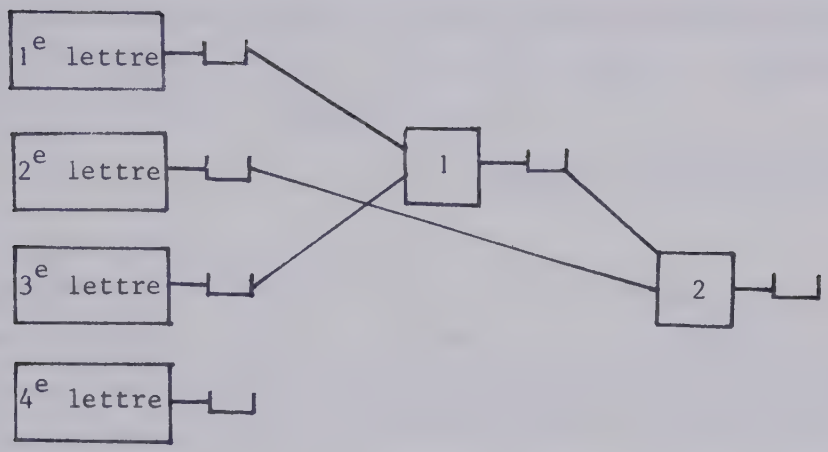
0	1	2	3	4	5	6	7	8	9	10
non	non	non	oui	oui	oui	oui	oui	oui	oui	oui



3. Le paragraphe I.2.1 montre de la même façon des opérateurs de différents niveaux qui travaillent sur des données qui sont les résultats des opérateurs du niveau précédent.

L'étage de reconnaissance de mots à partir des lettres peut aussi être décomposé. Soit par exemple à reconnaître les 4 mots suivants (voir [66]).

KIWI, ACCORD, ABIME, AMI



les tables des opérations 1 et 2 seront :

3e lettre \ 1e lettre	A	K
W	A.W	K.W
C	A.C	K.C
I	A.I	K.I

Opération 1

Opération 2

2e lettre \ attribut de O_1	A.W	K.W	A.C	K.C	A.I	K.I
I	inconnu	KIWI	inconnu	inconnu	inconnu	inconnu
C	inconnu	inconnu	ACCORD	inconnu	inconnu	inconnu
B	inconnu	inconnu	inconnu	inconnu	ABIME	inconnu
M	inconnu	inconnu	inconnu	inconnu	AMI	inconnu

Bien sûr dans cet exemple un opérateur suffisant est celui qui teste simplement la 2ème lettre ; son résultat permet de discriminer les 4 mots. L'exemple a été proposé au simple but d'explication et a été tiré d'un problème plus important que nous avons étudié en [66] et qui traitait la reconnaissance de 100 mots.

I.2.3. Niveaux différents d'opérateurs

Nous avons vu au paragraphe I.2.1 qu'un problème de reconnaissance de forme peut être décomposé en un ensemble de plusieurs étages de reconnaissance de formes primitives. Ces formes primitives sont des formes élémentaires dont l'existence est caractéristique de l'existence de la forme finale à reconnaître. Dans certains cas, en particulier le cas de la reconnaissance de mots, ces formes primitives existent naturellement et l'homme peut donner un nom à chacune d'elles (points, segments, angles, lettres...) ; l'homme a aussi une certaine conscience de ce qui est caractéristique a chaque niveau des formes du niveau supérieur : la conception de systèmes de reconnaissance est alors simplifiée.

Dans d'autres cas, (reconnaissance vocale, reconnaissance de forêts sur photos aériennes...) l'existence de primitives pose un problème et s'il en existe du moins sont elles inconnues.

Le paragraphe I.2.2. nous a montré que même à l'intérieur de chaque niveau, l'opérateur correspondant était une structure d'opérateurs plus simples. Ces opérateurs travaillent de la même façon à plusieurs niveaux les uns sur les résultats de ceux du niveau précédent.

Même dans les cas où l'existence de niveaux différents de primitives est inconnue à l'homme, si on peut concevoir un système de reconnaissance, la structure d'opérateurs à différents niveaux existera toujours.

On pourra simuler le système sur ordinateur. Le programme sera alors formé d'une succession d'instructions-machine qui sont effectivement des opérateurs au sens des définitions précédentes. Ces opérateurs travaillent sur des variables (registre ou mémoire) dont les valeurs ont été déterminées par les opérateurs précédents ou bien par les capteurs.

La seule exception est le programme avec boucles et saut conditionnel. Cette richesse des ordinateurs peut être utilisée avec profit dans les systèmes de reconnaissance des formes. Se rapporter au chapitre suivant sur les opérateurs de commande. C'est ce qui permet l'invariance à certaines transformations et diminue le nombre d'opérateurs nécessaires en utilisant les mêmes opérateurs sur des données différentes.

I.3. OPERATEURS SYNTAXIQUES

Une différence apparaît comme fondamentale entre les deux opérateurs décrits en I.2.3., les filtres linéaires et les opérateurs reconnaissant les mots.

En effet, les filtres linéaires peuvent être qualifiés d'opérateurs arithmétiques, les opérations élémentaires correspondantes étant des opérations somme, seuil ou multiplication.

Par contre, les opérateurs qui reconnaissent les mots ne travaillent plus sur des données représentant des valeurs numériques mais des valeurs syntaxiques. Chaque attribut prend en effet ses valeurs dans un alphabet de noms de formes. Dans ce cas comme dans l'autre, les opérations correspondantes peuvent être représentées par des tables de correspondance.

Ce sont des opérateurs syntaxiques.

Nous avons étudié certains algorithmes de reconnaissance syntaxique sur la reconnaissance d'une constellation de points indépendamment des rotations, translations, homothétie et de certaines déformations : [83],[82], et le chapitre suivant.

Les règles de récriture sont des exemples de tels opérateurs dans le cadre très rigide des grammaires formelles introduites par CHOMSKY.

Certains chercheurs ont essayé d'appliquer cette formalisation à la reconnaissance des formes : surtout KIRSCH [37], NARASHIMAN [52], certains en la modifiant un peu pour l'appliquer aux langages bidimensionnels :

ROZENFELD et PFALZ [55]. Voir aussi F. DACEY qui étudie en [12] une grammaire générant des triangles. Certains y introduisent des probabilités comme SWAIN et FU [87]. Enfin, SHAW [73] et EVANS [17] utilisent aussi le formalisme des grammaires formelles : le vocabulaire non terminal n'est pas constitué des éléments de la rétine de départ, mais de l'ensemble des concepts des formes reconnues. Ils tiennent compte aussi du phénomène bidimensionnel en utilisant des règles de liaison (voir les opérateurs de commande du chapitre II) généralisant les règles de concaténation introduites par la formalisation des monoïdes libres et des grammaires de Chomsky. Voir aussi les bilangages de PAIR et QUERE (1970).

I.4. FORMALISATION DES OPERATEURS ET STRUCTURES D'OPERATEURS

I.4.1. Définition formelle des opérateurs

On définit un opérateur de reconnaissance de formes par le 5 uple :

$$O = \{\{A_e\}, \{A_s\}, T_{es}, A_c, a_c\}$$

- A_e, A_s, A_c sont des variables : à une variable A correspond un ensemble d'états possibles ou valeurs a(k) en nombre fini. On notera :

$$A = \{a(k)\}$$

- $\{A_e\}$ sont les variables d'entrée
- $\{A_s\}$ sont les variables de sortie
- A_c est la variable d'activation : $A_c \in \{A_e\}$
- a_c est l'état d'activation de A_c
- T_{es} est l'opération associée à O. C'est la representation concrète d'une application de l'ensemble des états de A_e dans l'ensemble des états de A_s, ce peut être la table de correspondance de cette application.

Le temps est une variable particulière

$$T = \{t_1, t_2, \ldots, t_n\}$$

dont les états peuvent être mis en correspondance biunivoque avec des valeurs numériques toutes différentes. Ces états sont ordonnés suivant ces valeurs numériques croissantes.

A toute valeur de T, toute variable A est dans un état a(k) et un seul.

Lorsqu'un opérateur O est "activé", à l'instant t_i, les variables de sortie de O sont à l'instant t_{i+1} dans l'état qui est fonction de celui des variables d'entrée à l'instant t_i par la table de correspondance T_{es} de O.

Tout opérateur O est activé (ou actif) en t_i, si à t_i la variable A_c est dans l'état a_c.

Si à l'instant t_i, une variable V_1 est dans un certain état $a_1(k)$ et si à cet instant aucun opérateur ayant V_1 comme variable de sortie, n'a été activé, alors à t_{i+1} V_1 est encore dans l'état $a_1(k)$.

Chaque variable peut avoir un nom, de même chaque état de variable et, chaque table de correspondance.

On désigne alors un opérateur par une suite de symboles tels que les suivants :

$$| \text{ ACTI = 1 } | < [B] \leftarrow \text{TAB } [A1, A2] >$$

ACTI, B, A1, A2 sont des noms de variables. TAB est le nom d'un tableau de correspondance : il représente une opération.

La suite de symboles précédente signifie que quel que soit le temps T, si la variable ACTI a comme valeur 1, la variable B prend un état fonction des états de A_1 et A_2 par l'opération TAB.

I.4.2. Définition formelle des structures d'opérateurs

On définit une structure de reconnaissance de formes au sens large comme le quadruplet :

$$S_r = \{\{O_j \mid j = 1, n\}, G, A_O, A_V\}$$

- $\{O_j \mid j = 1, n\}$ est un ensemble fini de n opérateurs de reconnaissance.
- G est un sous graphe orienté dont le nombre de sommets est égal à n.
- A_O est une bijection de l'ensemble des sommets de G dans l'ensemble des opérateurs O_j. Nous appellerons par extension O_j le sommet correspondant à l'opérateur O_j par cette application.
- A_V est une application de l'ensemble des arcs de G dans l'ensemble des variables V des opérateurs O_j.

A_V est telle que sa restriction aux arcs incidents à un sommet quelconque O_j peut se décomposer en une application de l'ensemble des arcs incidents intérieurement à O_j dans l'ensemble des variables d'entrée de l'opérateur O_j et en une application de l'ensemble des arcs incidents extérieurement à O_j dans l'ensemble des variables de sortie de l'opérateur O_j.

On définit les variables d'entrée de la structure S_r comme les variables V_l telles que les arcs $A_V^{-1} \cdot V_l$ sont tous incidents intérieurement au sous graphe G.

On définit les variables de sortie de la structure S_r comme les variables V_s telles que les arcs $A_V^{-1} \cdot V_S$ sont tous incidents extérieurement au sous graphe G.

Les variables V_h de S_r telles qu'il existe un arc de $A_V^{-1} \cdot V_h$ incident au sous graphe G et qui ne sont ni variables de sortie ni variables d'entrée sont appelées variables hybrides.

Les variables extérieures de S_r sont les variables V_l, V_s et V_h.

Une condition d'existence d'une structure S_R est que quels que soient les états des variables d'entrée de S_R, il n'existe aucun temps t_i tel que deux opérateurs ayant une variable de sortie commune soient tous deux actifs.

Nous ne nous intéresserons dans la suite qu'à un sous ensemble des structures S_r :

Une structure de reconnaissance au sens restreint est une structure de reconnaissance S_R satisfaisant aux propriétés suivantes :

- la réunion de l'ensemble des variables d'entrée et de l'ensemble des variables hybrides de S_R est de cardinal supérieur à 1.
- la réunion de l'ensemble des variables de sortie et de l'ensemble des variables hybrides de S_R est non vide.
- Une seule des variables d'entrée est une variable d'activation. Elle peut avoir plusieurs états d'activation possibles correspondant à différents opérateurs internes de S_R.
- Lorsqu'au moins un opérateur de S_R est actif, on dit que S_R est active. A chaque état des variables d'entrée correspond un temps d'activation.

S_R est une structure de reconnaissance au sens restreint s'il existe une borne supérieure aux temps d'activation correspondant à tous les états des variables d'entrée.

Pendant l'activation de S_R aucune variable d'entrée ou hybride ne doit changer d'état par intervention d'opérateurs extérieurs, en particulier la variable d'activation. On peut utiliser pour ce faire des variables de sortie dont certains états correspondent à l'activation de S_R. Ces variables (sémaphores : cf. DIJSTRA [93]) peuvent servir d'entrée aux structures extérieures à S_R dont les variables de sortie sont les variables d'entrée de S_R (DIJSTRA [93]) ou à une structure plus complexe gérant les interactions entre les différentes structures (ARSAC [92]).

Une conséquence de ces hypothèses est qu'à chacun des états d'activation de S_R correspond une application de l'ensemble des états d'entrée de S_R dans l'ensemble des états des variables de sortie de S_R.

On peut donc faire correspondre à chaque état d'activation de S_R, un opérateur au sens du §I.4.1. qui a le même effet sur les variables de sortie de S_R, mais avec une durée d'activation différente et égale à l'unité. Cet opérateur est appelé opérateur associé à S_R par l'état d'activation correspondant.

I.4.3. Notation formalisée des structures de reconnaissance

Nous employons une notation dérivant à la fois de l'Algol et de la notation de Backus.

Un opérateur est défini par ses variables d'entrée et de sortie , et par son opération (table de correspondance) : exemple de DIFFER.

Une structure est définie par un enchaînement d'opérateurs élémentaires et de structures par l'intermédiaire de leurs variables d'activation : exemple de DISTEUCLID.

On retrouve ces notions d'opérateurs considérés comme structures d'opérateurs plus simples dans des publications concernant les formalisations des problèmes de microprogrammation. Surtout GLUSHKOW [21] mais aussi T. ITO [29] ,[30] ,[31] et SCHLAEPPI [70] . Celui-ci introduit un langage (LOTIS) qui peut décrire des structures d'opérateurs telles que celles que nous venons de définir. Mais dans le cas de la microprogrammation, la notion de structure d'opérateurs et de structure de structures à différents niveaux n'est sans doute pas encore aussi essentielle qu'en reconnaissance des formes.

Notation :

D → DISTEUCLID (X, Y, U, V)

Définition :

DEF [A → DISTEUCLID (B, C, D, E)]
DEBUT ;| VARACT = acti | < S_1 ← DIFFER (B, D) >,
< S_2 ← DIFFER (C, E) >, < VARACT ← act >
; | VARACT = act 2 | < S_1 ← CARRE (S_1) >, < S_2 ← CARRE (S_2) >, < VARACT ← act 3 > ;
| VARACT = act 3 | < A ← SOMME (S_1, S_2) > ; FIN
DEF [A ← DIFFER (B, C)]

DEBUT ; DIFFER ≡

	0	1	2	3
0	0	1	2	3
1	-1	0	1	2
2	-2	-1	0	1
3	-3	-2	-1	0

; FIN

etc...

C H A P I T R E I I

LES OPERATEURS DE COMMANDE

II.1. NOTION D'OPERATEUR DE COMMANDE

II.1.1. Notion de paramètre d'un opérateur

Un capteur visuel peut être composé :

- soit d'un damier de cellules photoélectriques disposées par exemple suivant un maillage carré. Chacune de ces cellules travaille en parallèle avec les autres. Il en vient un certain état de registre correspondant, suivant la lumière qu'elle recoit.

- soit d'un "flying spot". C'est un système électronique pouvant servir de périphérique à un ordinateur. Dans ce cas, l'ordinateur lui indique deux valeurs x et y correspondant à la position de l'image où le dispositif doit mesurer ponctuellement la lumière. Lors du positionnement en x et y le flying spot envoie à l'ordinateur une valeur discrète i correspondant à l'intensité de la lumière au point (x, y).

Le capteur visuel peut donc être considéré

- comme un opérateur donnant comme résultat la valeur de la matrice $\|i(x, y)\|$.

- comme un ensemble de n x m opérateurs numérotés par le couple (x, y), donnant chacun comme résultat i(x, y).

- comme un opérateur muni des paramètres (x, y) donnant comme attribut i(x, y).

Les deux premières définitions sont trop générales et par là-même moins riches que la troisième. En effet, celle-ci montre l'identité de chacun des opérateurs I(x, y) aux nombres x et y près.

Par ailleurs (on en verra des exemples) d'autres opérateurs peuvent utiliser les nombres x et y.

II.2.2. Les opérateurs de commande

La notion de paramètre provient de la mise en facteurs d'opérateurs ayant le même rôle et qui peuvent donc donner lieu au même cablage, ou au même programme (s'il s'agit d'un ordinateur). Le cablage ou le programme travaille alors différemment suivant les valeurs des paramètres correspondants.

Les opérateurs dits "de commande" permettront une utilisation pratique de ces paramètres.

II.2. EXEMPLES D'OPERATEURS DE COMMANDE

II.2.1. L'opérateur distance euclidienne

Considérons deux constellations de couples d'étoiles. Un capteur optique peut en donner après prétraitement les représentations suivantes :

Constellation A	Constellation B
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0	0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0	0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0

Figure 3

Ces deux représentations peuvent être les valeurs de deux matrices en ordinateur A(i, j) et B(i, j). Au vu de ces deux représentations, un être humain peut dire :

"Dans la constellation A les deux étoiles sont plus éloignées que dans la constellation B".

Il reconnaîtra A ou B en tenant compte de ce "fait" et ceci indépendamment de toute translation, rotation ou même déformation (dans une certaine mesure) des constellations.

Comment résoudre ce problème en ordinateur ?

Un ensemble d'opérateurs tels que ceux du chapitre I pourraient travailler sur les valeurs de la matrice de données. Mais il faudrait un nombre trop grand d'opérateurs pour résoudre le problème indépendamment de toute translation rotation et a fortiori déformation.

Par contre si le système est un ordinateur on peut concevoir un programme simple pour résoudre entièrement le problème :

1. Balayer la matrice pour trouver un point dont la valeur vaut 1
2. Noter ses coordonnées i_1, j_1
3. Balayer la matrice pour trouver un deuxième point dont la valeur vaut 1
4. Noter ses coordonnées i_2, j_2
5. Calculer la valeur de
$$D = \sqrt{(i_2 - i_1)^2 + (j_2 - j_1)^2}$$
6. Si $D > 5$ la constellation est A
7. Si $D \leqslant 5$ la constellation est B.

On remarque que les opérateurs des lignes 5, 6 et 7 sont des opérateurs au sens du chapitre I. Ce sont même des opérateurs arithmétiques.

Par contre, la recherche des valeurs de i_1, j_1, i_2, j_2 correspond à la notion nouvelle des opérateurs de commande. Dans cet exemple précis, l'opérateur de commande travaille sur les paramètres i et j de l'opérateur A(i, j) qui donne la luminosité en (i, j).

Cet opérateur est muni des deux **paramètres** i et j : au lieu d'avoir 9 x 10 = 90 opérateurs-capteurs différents, on n'en utilise qu'un seul A(i, j) mais paramètré par i et j.

Il est ainsi possible à un opérateur de travailler sur i et j pour commander l'opérateur A(i, j). Cette commande se fait ici par des opérateurs arithmétiques tels que i := i + 1 et des opérations seuil, ainsi que par des sauts conditionnels.

II.2.2. Autres exemples

Nous avons mis en oeuvre ([82],[83]) un algorithme de reconnaissance de constellations de plus de deux points. C'est un opérateur syntaxique mais sa puissance provient principalement du fait qu'il utilise des opérateurs de commande travaillant sur des qualificatifs. En effet, il consiste successivement en :

Opérateurs de commande :

1. Une recherche des points sur la rétine. A chaque nouveau point trouvé on affecte un nom (par exemple : $i \in \{1, 2, 3 \text{ etc...}\}$) et les coordonnées x_i et y_i du point.

2. On calcule (opérateur arithmétique) les valeurs :

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

3. Opérateur de commande : on trie l'ensemble des triplets $\{i, j, d_{ij}\}$ dans l'ordre des d_{ij} décroissants.

4. Opérateur syntaxique : la réponse de l'opérateur est le sous ensemble ainsi ordonné des doublets {i, j}. C'est un élément de vocabulaire construit sur l'alphabet des noms des points.

On constate ([82], [83]) que l'opérateur a des propriétés d'invariance par rotation, translation similitude et petites déformations.

Un autre exemple d'opérateur utilisant des opérateurs de commande est le programme d'extraction de squelette de formes bidimensionnelles étudié en [11] :

Contrairement aux autres algorithmes de recherche de squelette de la littérature actuelle, nous avons construit un algorithme simulant un processus de recherche faisant appel à des notions de formes élémentaires intuitives.

- notion de bras ou pseudopodes
- largeur de "bras"
- progression le long du bras
- extrémité de bras
- intérieur de courbe
- section droite d'un bras
- milieu d'une section droite d'un bras.

Certains opérateurs extraient les formes élémentaires (ou primitives). D'autres opérateurs font une recherche dirigée de positions où les opérateurs qui extraient les formes élémentaires vont travailler. Là encore on utilise une structure d'opérateurs arithmétiques, syntaxiques et de commande.

De la même façon nous avons utilisé [20] des opérateurs de commande dans le prétraitement d'image par filtrage digital linéaire, dans le but d'une reconnaissance de formes visuelles. Voir I.2.2.

En effet, les filtres utilisés extraient les traits lumineux de différentes orientations. A chaque orientation (θ) et chaque position (x, y) correspond un filtre $F_{\theta, x, y}$. On dit qu'un trait lumineux de direction θ existe en (x, y) si la valeur du resultat du filtre $F_{\theta, x, y}$ (opérateur arithmétique) dépasse un certain seuil s.

On peut utiliser ces opérateurs en les faisant travailler systématiquement suivant toutes leurs positions possibles et toutes leurs orientations. Mais une remarque intuitive permet de simplifier l'algorithme et de l'accélérer : la présence d'un trait lumineux à la position (x, y) et suivant la direction θ, est très correlée avec la présence d'un autre en (x + 1 cos θ, y + 1 sin θ) suivant la même direction ou une direction voisine. Ce fait est en rapport très étroit avec l'existence de la notion de ligne (succession de traits).

D'où la méthode utilisée de recherche de trait. On "parcourt" les lignes successivement en essayant par priorité les filtres de position voisine et de direction voisine : voir [80]. Cette méthode implique des opérateurs de commande travaillant sur les paramètres θ, x et y controlant séquentiellement les opérateurs arithmétiques que sont les filtres linéaires.

II.3. DEFINITION FORMELLE DES OPERATEURS DE COMMANDE

II.3.1. Introduction

La formalisation qui va suivre signifie en langage intuitif qu'un opérateur de commande est un opérateur (au sens du chapitre I) dont le but est de chercher en mémoire une variable ou un opérateur défini par son nom et ses paramètres. Nous allons définir en premier lieu les notions de variable matérielle, d'opérateur matériel, de structure matérielle d'opérateur, par la définition de la notion d'adresse associée.

L'utilisation des ordinateurs introduit la notion d'adresse par une numérotation binaire des mots de la mémoire de l'ordinateur. Une variable a une certaine adresse, de même qu'un programme et qu'un sous-programme. Dans ce dernier cas, l'ensemble des adresses est un ensemble ordonné isomorphe à l'ensemble des N premiers nombres entiers, N étant la capacité de la mémoire de l'ordinateur. Nous généralisons cette notion d'adresse.

II.3.2. Notions de variables matérielles, d'opérateurs matériels, de structures matérielles

Un support matériel d'opérateurs est représenté par un ensemble fini dénombrable $\Gamma = \{\alpha_j\}$, appelé ensemble des adresses α_j.

Une variable A_j est dite matérielle, s'il lui est associé biunivoquement une valeur $\alpha_j \in \Gamma$: α_j est l'adresse de la variable matérielle A_j.

Une opération T_{es} est dite matérielle, s'il lui est associé biunivoquement une valeur $\alpha \in \Gamma$. α est l'adresse de l'opération matérielle T_{es}.

Un opérateur est matériel si ses variables d'entrée et de sortie sont matérielles et que son opération associée est matérielle.

Une structure matérielle est formée de la même façon d'opérateurs matériels et de variables matérielles. Dans une même structure matérielle, la correspondance qui existe entre ces opérateurs et variables d'une part et leurs adresses d'autre part est binunivoque : à une adresse ne correspond qu'une variable ou qu'une opération.

II.3.3. Les opérateurs de commande

Considérons un ensemble de variables Γ_σ et un ensemble de variables Γ_a. Les variables Γ_σ seront appelées compteurs ordinaux, les variables Γ_a seront appelées registres d'adresses.

Dans le support matériel que constitue un ordinateur classique, il existe un seul compteur ordinal et un seul registre d'adresse. De manière générale dans tout support matériel de processus séquentiels il n'existe qu'un seul compteur ordinal.

Chaque état des Γ_σ et des Γ_a est mis en correspondance biunivoque avec une adresse $\alpha_j \in \Gamma$.

Un opérateur ne peut être actionné que si les adresses des variables d'entrée (dont la variable d'activation) et de sortie sont celles qui correspondent aux états des registres d'adresses. Il faut donc qu'il y ait au moins autant de registres d'adresses que de variables à chacun des opérateurs de la structure.

Un opérateur ne peut être actionné que si l'adresse de son opération correspond à l'un des états des compteurs ordinaux. Il ne peut donc y avoir plus d'opérateurs travaillant en parallèle que de compteurs ordinaux.

Un opérateur de commande est un opérateur O_j dont une au moins des variables de sortie est l'une des variables $\ulcorner_\sigma$ ou $\ulcorner_a$.

Reprenons la notation $S_1 \leftarrow$ DIFFER (B, D) de la page 34 , c'est la représentation d'un opérateur qui fait "la différence" entre les variables B et D.

On peut considérer ces symboles comme les états des variables d'entrée des opérateurs de commande de l'opérateur O_j. Les états de ses variables de sortie sont les adresses

1. des variables dont le "nom" est S_1, B et D
2. de l'opération dont le "nom" est DIFFER.

Nous devons donc maintenant considérer les noms des variables et des opérations comme les états de variables correspondant à des opérateurs de commande ; de la même façon, les "indices de variables". En effet, la notation

A(X, Y)

représente, "si X = 3 et Y = 4, l'élément de la matrice A, appartenant à la 3ème ligne et à la 4ème colonne". En fait, cette notation n'est utilisée que s'il existe plus ou moins explicitement un opérateur travaillant sur cette variable A(X, Y). A cette variable est alors associé un opérateur de commande admettant 3 variables d'entrée, l'un dans l'état "A", les deux autres qui sont simplement X et Y.

Lorsqu'un interpréteur doit décoder la suite de symbole A(X, Y), il agit comme un opérateur de commande. L'adresse de la variable codée A(X, Y) est souvent calculée comme étantl'adresse de A, additionnée de la valeur de X et de λ fois la valeur de Y.

Reprenons maintenant la notation de la page , des filtres linéaires utilisés dans [80] pour le prétraitement d'images. Ils sont notés :

$$^{O}_{I,\ J,\ \theta}$$

Ces opérateurs peuvent être représentés par la suite de symboles :

$$B \leftarrow F_{\theta}(A(I,\ J))$$

A(I, J) représente l'ensemble des valeurs des points de la rétine dans un carré centré en (I, J).

F_{θ} représente le filtre utilisé de direction θ.

La suite des symboles A(I, J) a le même sens que précédemment. Elle représente un état (A) de l'une des variables d'entrée d'un opérateur de commande et deux autres de ces variables (I et J).

Les symboles F_{θ} représentent, de même, l'état de la variable d'entrée de l'opérateur de commande qui provoque le filtrage, et la variable θ spécifiant le filtre à utiliser.

Les structures pratiques utilisées en programmation sont en général très complexes et font appel de façon importante à ces opérateurs de commande : procédures de tri, ... Les problèmes de reconnaissance des formes nécessitent aussi la présence de ces opérateurs (cf [83]).

Le support matériel que constituent les ordinateurs permet leur utilisation de manière très souple : c'est ce qui fait la richesse de ces outils qui ne sont pourtant utilisés la plupart du temps que comme manipulateurs de "fichiers". L'introduction des grammaires formelles si elle a permis de voir plus clair dans certains problèmes mettant en jeu des monoïdes libres, n'a que peu tenu compte de ces opérateurs de commande. En effet, dans une grammaire formelle les opérateurs sont les règles de récriture qu'elle spécifie. Les opérateurs de commande ne sont pas déterminés et tous peuvent être utilisés.

Nous rejoignons ici les idées émises par WINOGRAD en 71 [91] qui a étudié la compréhension par ordinateur de phrases en anglais courant, dans un certain univers sémantique (celui du "robot" du Massachusetts - Institute of Technology). Il écrit dans sa thèse :

" Il est paradoxal que les linguistes familiers avec les ordinateurs n'ont pas en général apprécié l'importance de l'aspect "opérateur de commande" en programmation ("control" aspect of programming) et n'ont pas utilisé les possibilités de description de processus pour leurs théories d'analyse de données. Ils se sont restreints à la place à des règles et des transformations des plus simples...comme si un programmeur devait rester attaché à des modèles aussi simples que les machines de Turing ou les post productions".

Nous verrons au chapitre IV une utilisation de certains opérateurs de commande, dans des problèmes de reconnaissance de formes. Il s'agira de faire travailler séquentiellement des opérateurs O_j de reconnaissance dans le but d'optimiser cette reconnaissance.

Dans les chapitres suivants où nous traitons de génération automatique d'opérateurs de reconnaissance, les opérateurs de commande n'occupent plus de place prépondérante. L'opérateur de commande utilisé est celui qui existe dans tous les ordinateurs classiques, il fait augmenter le compteur ordinal Γ_0 d'une unité chaque fois qu'un opérateur est activé.

Le problème de la génération des opérateurs de commande dans le cas général ne sera pas abordé. Il est pourtant fondamental dans les problèmes pratiques de recherche en mémoire, par exemple les mémoires d'ordinateur.

CHAPITRE III

INFORMATION UTILE APPORTEE PAR UN OPERATEUR

III.1. L'OPERATEUR IDEAL

Dans les problèmes pratiques de reconnaissance, les formes sont en général connues d'avance par l'homme, les conditions du problème sont assez bien définies et le but est de trouver un système automatique qui approche le plus possible la décision de l'homme.

On a donc à approcher un opérateur idéal O_{id} dont on ne connaît pas la structure mais seulement les résultats sur un ensemble de données.

Nous nous proposons dans ce chapitre de mesurer le degré de ressemblance entre un opérateur O_j donné et l'opérateur O_{id}.

III.2. PROBABILITES ET THEOREME DE BAYES

Appelons $\{\omega(i)\}$ l'ensemble des formes à reconnaître, ce sont les réponses de l'opérateur O_{id} et posons $\Omega = \{\omega(i) \mid i \in I\}$.

Associons à chaque $\omega(i)$ une probabilité a priori $p[\omega(i)]$, c'est la probabilité a priori d'avoir la forme $\omega(i)$ sans rien encore connaître sur la donnée à analyser.

Supposons que sur une certaine donnée, l'opérateur O_j donne le résultat $a_j(k) \in A_j$: on peut de la même manière définir $p[a_j(k)]$, mais aussi

$$p[\omega(i), a_j(k)]$$

$$p[a_j(k) \mid \omega(i)]$$

$$p[\omega(i) \mid a_j(k)]$$

Nous avons entre ces nombres les relations suivantes :

$$p[\omega(i) \mid a_j(k)] = \frac{p[\omega(i),\ a_j(k)]}{p[a_j(k)]}$$

$$p[\omega(i) \mid a_j(k)] = \frac{p[\omega(i)] \cdot p[a_j(k) \mid \omega(i)]}{p[a_j(k)]}$$

enfin

(1) $$\boxed{p[\omega(i) \mid a_j(k)] = p[\omega(i)] \cdot \frac{p[a_j(k) \mid \omega(i)]}{\sum_i p[\omega(i)] \cdot p[a_j(k) \mid \omega(i)]}}$$

C'est la relation de Bayes qui exprime le théorème de Bayes.

III.3. NEGUENTROPIE ET INFORMATION MUTUELLE

La probabilité $p[\omega(i) \mid a_j(k)]$ représente la probabilité a posteriori de la forme ω_i, après que l'expérience O_j ait donné le résultat $a_j(k)$.

Si nous trouvons un résultat $a_j(k)$ tel que les valeurs $p[\omega(i) \mid a_j(k)]$ soient égales à 0 pour toutes les valeurs de i sauf l'une d'elle i_1 pour laquelle :

$$p[\omega(i) \mid a_j(k)] = 1,$$

dans ce cas, la probabilité a posteriori d'avoir la forme $\omega(i_1)$ est 1. L'opérateur O_j a donc extrait la forme $\omega(i_1)$ de la donnée considérée.

Si nous trouvons un résultat $a_j(k)$ tel que pour toutes les valeurs de i :

$$p[\omega(i) \mid a_j(k)] = p[\omega(i)],$$

l'état des probabilités n'a pas changé et on dit intuitivement que l'opérateur O_j n'a apporté aucune information.

On appelle neguentropie de la "situation probabiliste"définie par le vecteur $\{p[\omega(i)]\}$ tel que $\sum_i p(\omega_i) = 1$ la quantité :

$$H[\Omega] = - \sum_i p[\omega(i)] \, . \, \text{Log } p[\omega(i)]$$

On démontre simplement qu'une condition nécessaire et suffisante pour que $H[\Omega]$ soit nulle est qu'il existe un i tel que $p[\omega(i)] = 1$.

La quantité $H(\Omega)$ a un certain nombre de propriétés étudiées en particulier dans la théorie des questionnaires, PICARD [56] et [57]. On y démontre en effet que la neguentropie représente le minimum du nombre moyen de questions à poser pour que l'une des formes $\omega(i)$ ait une probabilité de valeur 1. Si le logarithme est de base b, les questions à considérer sont les questions à b réponses.

Avant l'expérience O_j elle vaut donc :

$$H[\Omega] = - \sum_i p[\omega(i)] \, . \, \text{Log } p[\omega(i)]$$

Après l'expérience donnant le résultat $a_j(k)$, elle est remplacée par la néguentropie conditionnelle :

$$H[\Omega \mid a_j(k)] = - \sum_i p[\omega(i) \mid a_j(k)] \, . \, \text{Log } p[\omega(i) \mid a_j(k)]$$

Nous pouvons donc mesurer l'"information utile" apportée par la réponse $a_j(k)$ de l'opérateur O_j, par :

$$I[a_j(k) \; ; \; \Omega] = H[\Omega] - H[\Omega \mid a_j(k)]$$

En passant à la moyenne par rapport à k sur tous les $a_j(k)$, nous obtenons l'information mutuelle moyenne entre A_j et Ω :

$$(2) \qquad I[A_j ; \Omega] = \sum_{i,k} p[\omega(i), a_j(k)] . \text{Log} \frac{p[\omega(i), a_j(k)]}{p[\omega(i)] . p[a_j(k)]}$$

Démonstration de (2)

On écrit :

$$I[a_j(k) ; \Omega] = H(\Omega) - H[\Omega \mid a_j(k)]$$

$$= - \sum p[\omega(i)] \text{ Log } p[\omega(i)] + \sum_i p[\omega(i) \mid a_j(k)] \text{Log } p[\omega(i) \mid a_j(k)]$$

et $$I[A_j ; \Omega] = H(\Omega) - \sum_k p[a_j(k)] \; H[\Omega \mid a_j(k)]$$

Or $$H(\Omega) = H(\Omega) \sum_k p[a_j(k) \mid \omega(i)]$$

Car $$\sum_k p[a_j(k) \mid \omega(i)] = 1$$

$$I[A_j \quad ; \Omega] = - \sum_{i,k} p[a_j(k) \mid \omega(i)]\ p[\omega(i)]\ \log p[\omega(i)]$$

$$+ \sum_{i,k} p[a_j(k)]\ p[\omega(i) \mid a_j(k)]\ \log p[\omega(i) \mid a_j(k)]$$

Or $p[\omega(i), a_j(k)] = p[\omega(i)]\ p[a_j(k) \mid \omega(i)] = p[a_j(k)]\ p[\omega(i) \mid a_j(k)]$

$$I[A_j \quad ; \Omega] = \sum_{i,k} p]\omega(i), a_j(k)[\ \log \frac{p[\omega(i) \mid a_j(k)]}{p[\omega(i)]}$$

et finalement(2) Q. E. D.

Cette quantité $I[A_j ; \Omega]$ représente l'information au sens de SHANNON de l'opérateur O_j, considéré comme un canal d'information. Cette quantité a été proposée par LEWIS [41] (voir FU [20]). Nous l'avons utilisée pour estimer les opérateurs dans les problèmes de reconnaissance des formes. Souvent, nous l'appellerons aussi "information utile apportée par l'opérateur considéré dans le but de la reconnaissance des formes Ω".

Cette information utile dans le but de reconnaître Ω doit être distinguée de l'information utile introduite par BELIS et GUIASU (voir [57])

Les auteurs : KHINCHIN [36], LEWIS [41], BENZECRI [3] définissent $H[A]$ par un ceratin nombre d'axiomes. En particulier, BENZECRI définit $H[A] = \sum_{a \in A} p[a]\ .\ H[a]$ de telle manière que :

1. H[a] est une fonction réelle positive et décroissante de p[a].

2. Si A et A' sont deux ensembles statistiquement indépendants :

$$H(A, A') = H(A) + H(A')$$

où $(A, A') = \{(a, a') \mid a \in A, a' \in A'\}$

3. Si $p[a] = \frac{1}{2}$, $H[a] = 1$.

BENZECRI démontre dans ce cas, que la seule fonction répondant à ces hypothèses est

$$H[A] = - \sum_{a \in A} p[a] \cdot \text{Log } p[a]$$

FANO [8] et BENZECRI [3] démontrent que $I[A_j ; \Omega]$ est une quantité positive ou nulle dans tous les cas. Par contre, $I[a_j(k) ; \Omega]$ dont l'espérance mathématique est positive ou nulle peut pour certaines valeurs de k être négative (BENZECRI [3]).

Comme $H[\Omega \mid A_j] = \sum_k p(a_j(k)\ H[\Omega \mid a_j(k)]$ est positive :

(3) $$\boxed{0 \leqslant I[\Omega\ ;\ A_j] \leqslant H[\Omega]}$$

Lorsque le maximum $H[\Omega]$ de $I[\Omega\ ;\ A_j]$ est atteint : $H[\Omega \mid A_j] = 0$,
Alors quel que soit $a_j(k)$;

$$H[\Omega \mid a_j(k)] = 0$$

Dans ce cas, quelle que soit la réponse $a_j(k)$, il existe une forme $\omega(i)$ telle que $p[\omega(i) \mid a_j(k)] = 1$; il y a alors une correspondance univoque entre les réponses de O_j et les formes $\omega(i)$. O_j reconnaît donc les formes sans erreur.

On démontre aussi (voir [3],[18],[36]) les relations suivantes :

(4) $$I[Y\ ;\ X] = I[X\ ;\ Y] = H[Y] - H[Y\ /\ X] = H[X] - H[X\ /\ Y] \geqslant 0$$

(5) $$H[X,\ Y] = H[X] + H[Y\ /\ X] = H[Y] + H[X\ /\ Y]$$

(6) D'où : $$I[X\ ;\ Y] = H[X] + H[Y] - H[X,\ Y]$$

(7) Aussi : $$I[X\ ;\ YZ] = I[X\ ;\ Y] + I[X\ ;\ Z\ /\ Y]$$

De même, si l'ensemble probabilité X a M éléments :

$$H[X] \leqslant \text{Log } M$$

L'égalité a lieu si et seulement si les M éléments de X sont équiprobables.

Etant donné l'opérateur O_j utilisé pour la reconnaissance des formes Ω, la relation :

(8) $$I[A_j\ ;\ \Omega] = H[A_j] - H[A_j \mid \Omega]$$

signifie que la valeur de l'entropie conditionnelle $H[A_j \mid \Omega]$ est la quantité dont il faut diminuer l'information globale $H[A_j]$ apportée par O_j pour obtenir l'information utile $I[A_j ; \Omega]$.

Nous pouvons appeler cette quantité "information inutile apportée par O_j, dans le but de la reconnaissance des formes Ω". Elle peut être rapprochée du concept d'information esthétique en acoustique introduite par LEIPP : voir LIENARD [43].

III.4. MESURE DE LA QUALITE DES CARACTERISTIQUES EN RECONNAISSANCE DES FORMES

Ainsi que nous l'avons vu au chapitre I, une forme peut être mise en correspondance biunivoque avec un résultat d'opérateur dans un système de reconnaissance de forme. Il y a aussi correspondance entre toute caractéristique d'une forme donnée et l'un des résultats intermédiaires des opérateurs dont l'assemblage constitue le système de reconnaissance de la forme donnée parmi d'autres.

On peut donc mesurer d'après les résultats du paragraphe précédent la qualité de la caractéristique $a_j(k)$ d'une forme $\omega(i_1)$ à reconnaître parmi les formes $\{\omega(i)\}$.

Nous utiliserons la quantité :

$$I[a_j(k) ; \Omega]$$

On remarque que la notion de forme caractéristique dépend dans ce cas non seulement de $\omega(i)$ mais du contexte $\{\omega_i \mid i \neq i_1\}$ de $\omega(i_1)$.

On mesure la qualité d'un ensemble de formes caractéristiques $A_j = \{a_j(k)\}$, d'un ensemble de formes $\Omega = \{\omega_i\}$ à reconnaître, par la quantité :

$$I[A_j ; \Omega]$$

LEWIS semble être le premier à avoir utilisé cette fonction en reconnaissance des formes [41]. Il justifie son emploi par le fait qu'elle est une fonction monotone du pourcentage de bonne reconnaissance et qu'elle est additive sur un groupe de caractéristiques.

Le numéro spécial de Pattern Recognition d'avril 71 sur l'extraction de caractéristiques contient une étude des algorithmes d'extraction de caractéristiques et des différentes fonctions d'évaluation se trouvant dans la littérature. S. KAMAL ABDALI [34]. Dans ce même numéro E. RASEK [58] étudie de manière plus précise la fonction information mutuelle introduite en reconnaissance des formes par LEWIS et que nous utilisons.

KOVALEVSKY utilise aussi cette quantité dans [38] par le biais de $H[\Omega \mid A_j]$ et étudie le lien entre l'erreur donnée par l'algorithme et la valeur de $H[\Omega \mid A_j]$ (voir III.6).

BONGARD décrit aussi cette fonction par l'étude des travaux de SHANNON. Mis à part LEWIS qui a utilisé cette fonction dans un procédé séquentiel de reconnaissance de caractères en majuscules-baton manuscrits, les études ont surtout revétu un caractère théorique sans utilisation dans les problèmes pratiques (voir travaux de SHANNON et PICINBONO).

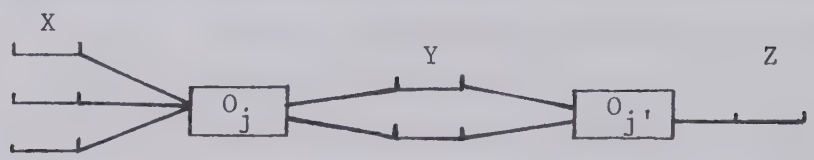
III.5. MISE EN SERIE ET EN PARALLELE D'OPERATEURS

III.5.1. Mise en série d'opérateurs

Au paragraphe I.2.3., nous avons décrit à l'aide de différents exemples comment les opérateurs de reconnaissances peuvent être décomposés en structures d'opérateurs plus simples à plusieurs niveaux. Nous allons étudier comment évolue l'information mutuelle $I[A_j ; \Omega]$, en fonction des niveaux des opérateurs O_j (voir [3], [18], [36]).

Théorème 1

Soient deux opérateurs en série O_j et O'_j :



Des valeurs $Y = \{y_j\}$ sont fonctions par O_j de $X = \{x_l\}$

$$Y = f_{O_j}[X]$$

et des valeurs $Z = \{z_k\}$, fonctions par O_j, des $Y = \{y_j\}$

$$Z = f_{O_j}[Y]$$

Alors, quel que soit l'ensemble $\Omega = \{\omega(i)\}$ si l'ensemble $\{Y, Z, \Omega\}$ est probabilisé :

(9) $$\boxed{I[\Omega ; Z] \leqslant I[\Omega ; Y]}$$

Démonstration (voir BENZECRI [3] et KHINCHIN [36]).

1°) Démontrons d'abord que la fonction $y = x \text{ Log } x$ est une fonction convexe pour $0 \leqslant x \leqslant 1$, c'est-à-dire que quel que soit l'ensemble $\{a_j \mid a_j \geqslant 0 ; \sum_j a_j = 1\}$,

$$-\left(\sum_j a_j x_j\right) \text{Log} \left(\sum_j a_j x_j\right) \geqslant -\sum_j a_j (x_j \text{ Log } x_j)$$

En effet, cette inégalité se transforme successivement en :

$$\sum_j a_j\, x_j\, [\mathrm{Log}(\sum_k a_k\, x_k) - \mathrm{Log}\, x_j] \leqslant 0$$

$$\sum_j a_j\, x_j\, (\mathrm{Log}\, \frac{\sum_k a_k\, x_k}{x_j}) \leqslant 0$$

Or :

$$\mathrm{Log}\, U \leqslant U - 1$$

Donc :

$$\mathrm{Log}\, \frac{\sum_k a_k\, x_k}{x_j} \leqslant \frac{\sum_k a_k (x_k - x_j)}{x_j}$$

(puisque $\sum_k a_k = 1$).

Alors :

$$\sum_j a_j\, x_j (\mathrm{Log}\, \frac{\sum_k a_k\, x_k)}{x_j}) \leqslant \sum_j a_j \sum_k a_k (x_k - x_j) \leqslant (\sum_j a_j \sum_k a_k\, x_k) - (\sum_k a_k \sum_j a_j\, x_j)$$

Cette dernière quantité étant nulle, l'inégalité du début est démontrée.

2°) Calculons dans les hypothèses du théorème 1, $H[\Omega \mid Y]$:

$$H[\Omega \mid Y] = - \sum_j p[y_j]\, [\sum_i p[\omega(i) \mid y_j]\, \mathrm{Log}\, p[\omega(i) \mid y_j]]$$

De même :

$$H[\Omega \mid Z] = - \sum_k p[z_k]\, [\sum_i p[\omega(i) \mid z_k]\, \mathrm{Log}\, p[\omega(i) \mid z_k]]$$

Or :

$$p[\omega(i) \mid z_k] = \frac{p[\omega(i), z_k]}{p[z_k]} = \frac{\sum_{j \in J_k} p[\omega(i), y_j]}{p[z_k]} = \sum_{j \in J_k} \frac{p[y_j]}{p[z_k]} p[\omega(i) \mid z_j]$$

(J_k est l'ensemble des j tels que $z_k = f_{O_j}, [y_j]$).

Donc :

$$H[\Omega \mid Z] = \sum_k p[z_1] \left[\sum_i \left[\sum_{j \in J_k} \frac{p[y_j]}{p[z_k]} p[\omega(i) \mid y_j]\right] \text{Log} \left[\sum_{j \in J_K} \frac{p[y_j]}{p[z_k]} p[\omega(i) \mid y_j]\right]\right]$$

Or la convexité de la fonction x. Log x entraîne (voir 1°))

$$- \left[\sum_{j \in J_K} \frac{p[y_j]}{p[z_k]} p[\omega(i) \mid y_i]\right] . \text{Log}\left[\sum_{j \in J_K} \frac{p[y_j]}{p[z_k]} p[\omega(i) \mid y_j]\right] \geqslant$$

$$- \sum_{j \in J_K} \frac{p[y_j]}{p[z_k]} p[\omega(i) \mid y_j] \text{ Log } p[\omega(i) \mid y_j]]$$

D'où :

$$H[\Omega \mid Z] \geqslant - \sum_j p(y_j) \left[\sum_i p[\omega(i) \mid y_j] \text{ Log } p[\omega(i) \mid y_j]\right]$$

C'est-à-dire : $H[\Omega \mid Z] \geqslant H[\Omega \mid Y]$

Comme : $I[\Omega ; Z] = H[\Omega] - H[\Omega \mid Z]$

On a bien l'inégalité annoncée sur les informations mutuelles.

Remarque :

On démontre aussi de la même façon que

(10) $$\boxed{H[Z] \leqslant H[Y]}$$

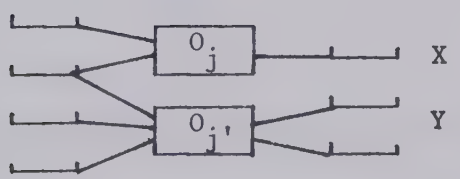
$$\text{et} \quad H[\Omega, Z] \leqslant H[\Omega, Y]$$

La première de ces deux inégalités signifie que l'information totale ne peut croître par un traitement de données, quel qu'il soit :

L'égalité $I[\Omega ; Z] \leqslant I[\Omega ; Y]$ montre qu'il en est de même de l'information utile.

III.5.2. Mise en parallèle d'opérateurs

Etudions l'information utile apportée par un ensemble de deux opérateurs mis en parallèle.



Théorème 2

Soient deux opérateurs O_j et $O_{j'}$ en parallèle dont les réponses sont $X = \{x_j\}$ et $Y = \{y_k\}$.

Quel que soit l'ensemble $\Omega = \{\omega(i)\}$ si l'ensemble $(X \otimes Y \otimes \Omega)$ est probabilisé :

(11) $$\boxed{I[\Omega ; X, Y] \geqslant I[\Omega ; X]}$$

en notant X, Y l'ensemble $\{x_j, y_k\}$.

Démonstration

Nous avons d'une part :

$$H[\Omega \mid X, Y] = - \sum_{j,\,k} p[x_j, y_k] \left[\sum_i p[\omega(i) \mid x_j, y_k] \operatorname{Log} p[\omega(i) \mid x_j, y_k]\right]$$

d'autre part :

$$H[\Omega \mid X] = - \sum_j p[x_j] \sum_i p[\omega(i) \mid x_j] \operatorname{Log} p[\omega(i) \mid x_j]]$$

$$= - \sum_j \left\{\left[\sum_k p[x_j, y_k]\right] \cdot \left[\sum_i \left[\sum_k p[\omega(i) \mid x_j, y_k]\right] \cdot \operatorname{Log} \sum_k p[\omega(i) \mid x_j, y_k]\right]\right\}$$

Cette valeur est supérieure ou égale à :

$$- \sum_j \left\{\left[\sum_k p(x_j, y_k)\right] \cdot \left[\sum_i \sum_k p[\omega(i) \mid x_j, y_k] \operatorname{Log} p[\omega(i) \mid x_j, y_k]\right]\right]$$

en vertu de la connexité de la fonction $x \operatorname{Log} x$ (paragraphe précédent) et a fortiori supérieure ou égale à :

$$- \sum_j \left\{\sum_k \left[p[x_j, y_k] \cdot \sum_i \left[p[\omega(i) \mid x_j, y_k] \operatorname{Log} p[\omega(i) \mid x_j, y_k]\right]\right]\right\}$$

qui est la valeur de $H[\Omega \mid X, Y]$

D'où :

$$H[\Omega \mid X, Y] \leqslant H[\Omega \mid X]$$

et $$I[\Omega ; X, Y] \geqslant I[\Omega ; X]$$

Remarque

De la même façon

(12) $$H[X, Y] \geqslant H[X]$$

Ces deux relations expriment l'idée intuitive que la considération de nouvelles données ne peut qu'augmenter tant l'information totale que l'information utile.

Théorème 3

Dans les mêmes hypothèses que le théorème 2, nous avons aussi la relation :

(13) $$\boxed{I[\Omega ; X, Y] \geqslant I[\Omega ; X] + I[\Omega ; Y] - I[X ; Y]}$$

Démonstration : successivement :

$$I[\Omega ; X, Y] = H[\Omega] + H[X, Y] - H[\Omega, X, Y]$$

$$= H[\Omega] + H[X] + H[Y] - I[X ; Y] - H[\Omega, X, Y]$$

$$= I[\Omega ; X] + H[Y] - I[X ; Y] + H[\Omega, X] - H[\Omega, X, Y]$$

$$I[\Omega ; X, Y] = I[\Omega ; X] - I[X ; Y] + I[Y ; X, \Omega]$$

Posons $$F = I[Y ; X, \Omega] - I[Y ; \Omega]$$

Alors :

$$I[\Omega\ ;\ X,\ Y] = I[\Omega\ ;\ X] + I[Y\ ;\ \Omega] - I[X\ ;\ Y] + F$$

Or F est positif, d'après le théorème 2, d'où l'inégalité (12).

Remarque :

D'après les relations (4) et (5), on a :

$$H[X,\ Y] = H[X] + H[Y\ /\ X] = H[X] + H[Y] - I[Y\ ;\ X]$$

Donc :

(14) $$H[X,\ Y] \leqslant H[X] + H[Y]$$

On a ici une borne supérieure de l'information totale apportée par O_j et $O_{j'}$. Il n'y a pas de relation équivalente à la relation (13) donnant une borne inférieure de l'information totale, en dehors de la relation (12) :

$$H[X,\ Y] \geqslant H[X]$$

Par contre, il n'existe pas de borne supérieure de $I[\Omega\ ;\ X,\ Y]$ inférieure à

$$\min \{H[\Omega],\ H[X,\ Y]\}$$

en particulier l'inégalité :

$$I[\Omega\ ;\ X,\ Y] \leqslant I[\Omega\ ;\ X] + I[\Omega\ ;\ Y]$$

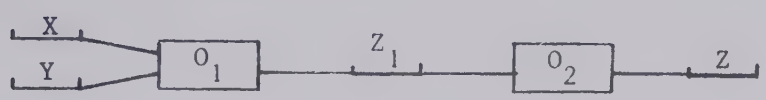
est fausse.

III.5.3. Exemple de l'addition

Soient deux variables X et Y pouvant prendre respectivement les valeurs $X = \{x_i\}$ et $Y = \{y_j\}$ et la fonction $Z = X + Y$ prenant les valeurs $Z = \{x_i + y_j\}$.

Si à X et Y correspondent $I[\Omega ; X]$ et $I[\Omega ; Y]$, peut on prévoir les valeurs de $I[\Omega ; Z]$?

Appliquons les résultats obtenus en décomposant l'opérateur addition en deux étages : les opérateurs O_1 et O_2.



Si X et Y ont 4 réponses chacun (0, 1, 2, 3) : l'addition est représentée par sa table :

	0	1	2	3
0	0	1	2	3
1	1	2	3	4
2	2	3	4	5
3	3	4	5	6

Supposons que au lieu d'avoir cette table nous ayons :

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

Cette table est la table de l'opération effectuée par O_1. Celle de l'opérateur O_2 est :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1	2	3	1	2	3	4	2	3	4	3	3	4	5	6

Les relations des paragraphes précédents donnent :

- opérateurs en parallèle :

$$I[\Omega ; Z_1] \geqslant I[\Omega ; X] + I[\Omega ; Y] - I[X ; Y]$$
$$\geqslant \max\{I[\Omega ; X], I[\Omega ; Y]\}$$

- Opérateurs en série :

$$I[\Omega ; Z] \leqslant I[\Omega ; Z_1]$$

On ne peut rien déduire de ces inégalités sur la valeur de $I[\Omega ; Z]$ par rapport à $I[\Omega ; X]$ et $I[\Omega ; I]$.

Le choix de l'opération effectuée (ici l'opération +) se traduit au niveau de O_2 par une plus ou moins grande diminution de l'information utile : ΔI.

Plus précisément, supposons que nous ayons, au lieu de la table d'addition, la table suivante :

	0	1	2	3
0	4	3	0	5
1	3	0	5	1
2	0	5	1	2
3	5	1	2	6

Nous avons ainsi changé les "noms" des réponses de l'opérateur addition suivant la correspondance :

0 → 4
1 → 3
2 → 0
3 → 5
4 → 1
5 → 2
6 → 6

Dans ce cas : l'information utile apportée par l'opérateur est inchangée.

L'opération effectuée n'a de l'importance vis-à-vis de la baisse ΔI de l'information utile que par le partage en classe qu'elle effectue sur les données traitées, et ce, indépendemment du nom affecté à chaque classe. Il y a donc accord entre la notion d'information utile et l'introduction des opérateurs en reconnaissance des formes du paragraphe I.1.4.

III.6. RELATION ENTRE LA PROBABILITE D'ERREUR ET L'INFORMATION UTILE

1°) Cas de deux formes

Supposons qu'après un certain nombre d'expériences A l'entropie conditionnelle soit voisine de zéro.

$$H[\Omega \mid A] = - p(\omega(1)) \text{ Log } p(\omega(1) - p(\omega(2)) \text{ Log } p(\omega(2)) \simeq 0$$

Ce fait entraîne que l'une (par exemple $p(\omega(1))$) des deux probabilités est voisine de 1. Si on prend la décision $\omega(1)$, la probabilité d'erreur est $p(\omega(2)) = \epsilon$.

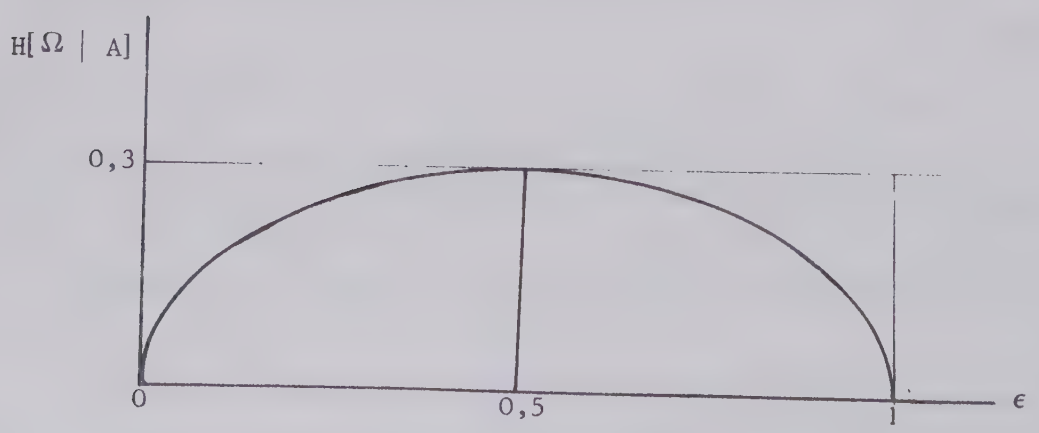
D'où la relation :

(15) $$H = - (1 - \epsilon) \text{ Log } (1 - \epsilon) - \epsilon \text{ Log } \epsilon$$

Si ϵ est petit :

(16) $$\boxed{H[\Omega \mid A] \sim \epsilon \text{ Log } \epsilon}$$

Si ϵ est non petit, la relation (13) est exprimée par le graphique suivant (les logarithmes décimaux ont été utilisés).



2°) Cas de plus de deux formes

Si l'une des probabilités, $p(\omega(1))$ par exemple, est supérieure à 0,5 et si on prend la décision $\omega(1)$, la probabilité d'erreur est $(1 - p(\omega(1)))$. Par contre, on ne sait rien sur l'entropie conditionnelle, seulement qu'elle a une valeur comprise entre 1°) l'entropie conditionnelle de la situation correspondant à

$$\{p(\omega(i)) = \frac{1 - p(\omega(i))}{n - 1} \mid \forall_i \neq 1\}$$ (équiprobabilité des autres formes)

et 2°) l'entropie conditionnelle de la situation :

$$\{\exists_i \neq 1,\ p(\omega(i)) = 1 - p(\omega(i)), \forall_j \neq 1 \text{ et } i,\ p(\omega(j)) = 0\}$$ (les autres formes sauf une ont une probabilité nulle)

D'où :

(17) $$\underline{- (1 - \epsilon) \text{ Log } (1 - \epsilon) - \epsilon \text{ Log } \epsilon \leqslant H \leqslant - (1 - \epsilon) \text{ Log } (1 - \epsilon)}$$

$$\underline{- \epsilon \text{ Log } \epsilon + \epsilon \text{ Log } (n - 1)}$$

Si ϵ est petit :

(18) $$\boxed{H[\Omega \mid A] \sim \epsilon \text{ Log } \epsilon}$$

On a la même relation que (16).

Si ϵ est non petit, la borne inférieure de $H[\Omega \mid A]$ suit la courbe donnée au 1°), la borne supérieure suit une courbe dépendant du nombre n :

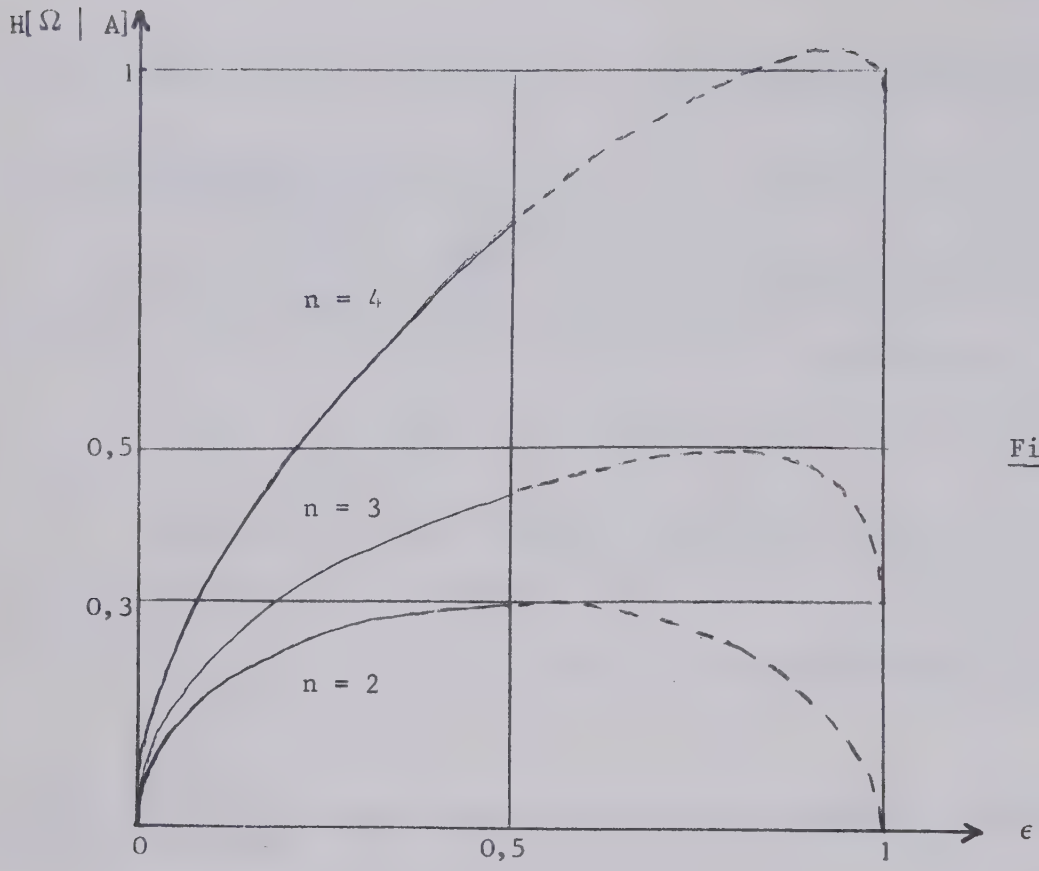


Figure 7

KOVALEVSKY [38] étudie de même les rapports entre les probabilités d'erreur et l'information utile et cite la démonstration par FEINSTEIN [19]. de la relation (17).

Les relations (16) et (18) montrent qu'au voisinage de $\epsilon = 0$, la néguentropie conditionnelle représente l'erreur en donnant d'autant plus de précision à la mesure de l'erreur, que l'erreur est faible.

Par ailleurs, l'avantage de la néguentropie conditionnelle, (ou de l'information mutuelle) pour la mesure des performances d'un opérateur, est qu'elle est définie dans tous les cas, même lorsque $H[\Omega \mid A]$ n'est pas voisin de zéro et où aucune décision ne peut être prise à moins d'un taux d'erreur inacceptable.

III.7. LA MESURE DES PROBABILITES ET DES FONCTIONS NEGUENTROPIE ET INFORMATION MUTUELLE

Dans toutes nos expériences pratiques, nous avons mesuré les probabilités par les fréquences d'apparition sur un ensemble important.

Nous avons établi deux ensembles, un ensemble d'apprentissage et un ensemble de test. Chacun d'eux comprend une liste de données, à chacune de celles-ci correspond le nom de la forme qu'elle représente, c'est-à-dire un des états $\{\omega(i)\}$ d'une variable Ω, introduit en ordinateur par l'homme (le "professeur" de l'algorithme d'apprentissage). Voir page 45.

Pendant la phase d'apprentissage nous déterminons les probabilités $p[\omega(i), a_j(k)]$, en parcourant l'ensemble d'apprentissage et en incrémentant la valeur $n[\omega(i), a_j(k)]$ correspondant à la conjonction de $\omega(i)$ et $a_j(k)$. Les probabilités s'obtiennent en divisant ces valeurs par le nombre total de données de l'ensemble d'apprentissage. La figure 13 donne un exemple de tels tableaux de probabilités dans le cas d'opérateurs utilisés en reconnaissance de la parole.

NILSSON (1965) expose de manière complète les méthodes de détermination des probabilités [53]. Une méthode simple est présentée par HATTAWAY, et AL [26].

MINSKY [49] détermine les probabilités par une suite de W_{ij}, prenant itérativement ses valeurs de la manière suivante : si l'évènement (ij) se produit, W_{ij} est remplacé par $(W_{ij} + 1)\theta$, sinon par $\theta.W_{ij}$: θ étant inférieur à 1.

Notre méthode est un cas particulier de celle-ci, en prenant θ égal à 1.

Le problème essentiel est la détermination du nombre minimal de données d'apprentissage pour obtenir des probabilités satisfaisantes, et surtout des valeurs de $I[O_i ; O_j]$ satisfaisantes. C'est le problème de l'<u>ergodicité</u> de l'ensemble d'apprentissage.

Nous avons fait des expériences à ce sujet en mesurant l'information mutuelle et l'entropie sur un ensemble de données d'apprentissage dont le nombre croit.

Il s'agissait de données vocales fournies par un vocoder;les formes Ω sont des phonèmes, les opérateurs O_1 et O_2 sont les canaux 3 et 4 du vocoder ; l'opérateur O_3 fait un certain regroupement des réponses des canaux 1 à 3, l'opérateur O_4 fait de même sur les canaux 4 à 6.

On voit sur la figure 8 que les valeurs des quantités $H[O_i]$ et $I[O_i ; O_j]$ se stabilisent,après un nombre de données de l'ordre de 100 : plus précisément :

Nombre de données	Valeur de H approchée à :	Valeur de I approchée à :
28	10 %	25 %
57	5 %	15 %
73	2 %	10 %
81	1 %	5 %
215	1 %	1 %
1284	0 %	0 %

Nous avons fait dans chacune de nos expériences des mesures semblables d'ergodicité.Dans tous les cas, nous avons une convergence de $I[A ; \Omega]$ après un nombre d'échantillons de l'ordre de 100.

Il semble y avoir une relation entre ce nombre d'échantillons et le nombre des formes Ω à reconnaître. Sans qu'aucune étude systématique n'ait été faite, le nombre d'échantillons d'apprentissage à prendre doit être de l'ordre de 2 à 3 fois le nombre de formes à reconnaître si on veut une valeur de $I[A ; \Omega]$ approchée à moins de 10 % et si le nombre des réponses A est de l'ordre de 10 à 16.

Dans les expériences décrites dans le chapitre suivant nous nous sommes contentés d'un nombre de 276 mesures ce qui donne donc une bonne précision sur les valeurs de H et I. Par contre, les probabilités d'erreur sont établies avec moins de précision, surtout les probabilités

conditionnelles d'erreur. En effet, $H[\Omega \mid A_j]$ est une moyenne de ces probabilités et l'erreur sur la moyenne est toujours plus faible que l'erreur sur chacun des termes.

Nous avons aussi étudié les variations de l'entropie et de l'information mutuelle d'opérateurs travaillant sur des données vocales en fonction du temps. Le but a été d'examiner les corrélations entre ces variations et l'existence de phonèmes (ou d'autres formes sonores).

La figure 9 montre les deux courbes d'évolution de $H_5[A_6]$ et $I_5[A_6 ; A_7]$ en fonction du temps :

- A_6 et A_7 sont les résultats des opérateurs canaux n° 6 et 7 du vocoder
- H_5 et I_5 sont calculés sur des ensembles de 5 données que l'on décale successivement pour obtenir les variations de H_5 et I_5 en fonction du temps.

Les courbes présentées figure 9 ont été obtenues lors de la prononciation du mot stratus devant le vocoder.

La courbe de l'Information mutuelle semble présenter des valeurs différentes de la moyenne (maxima et minima) aux emplacements approximatifs des phonèmes.

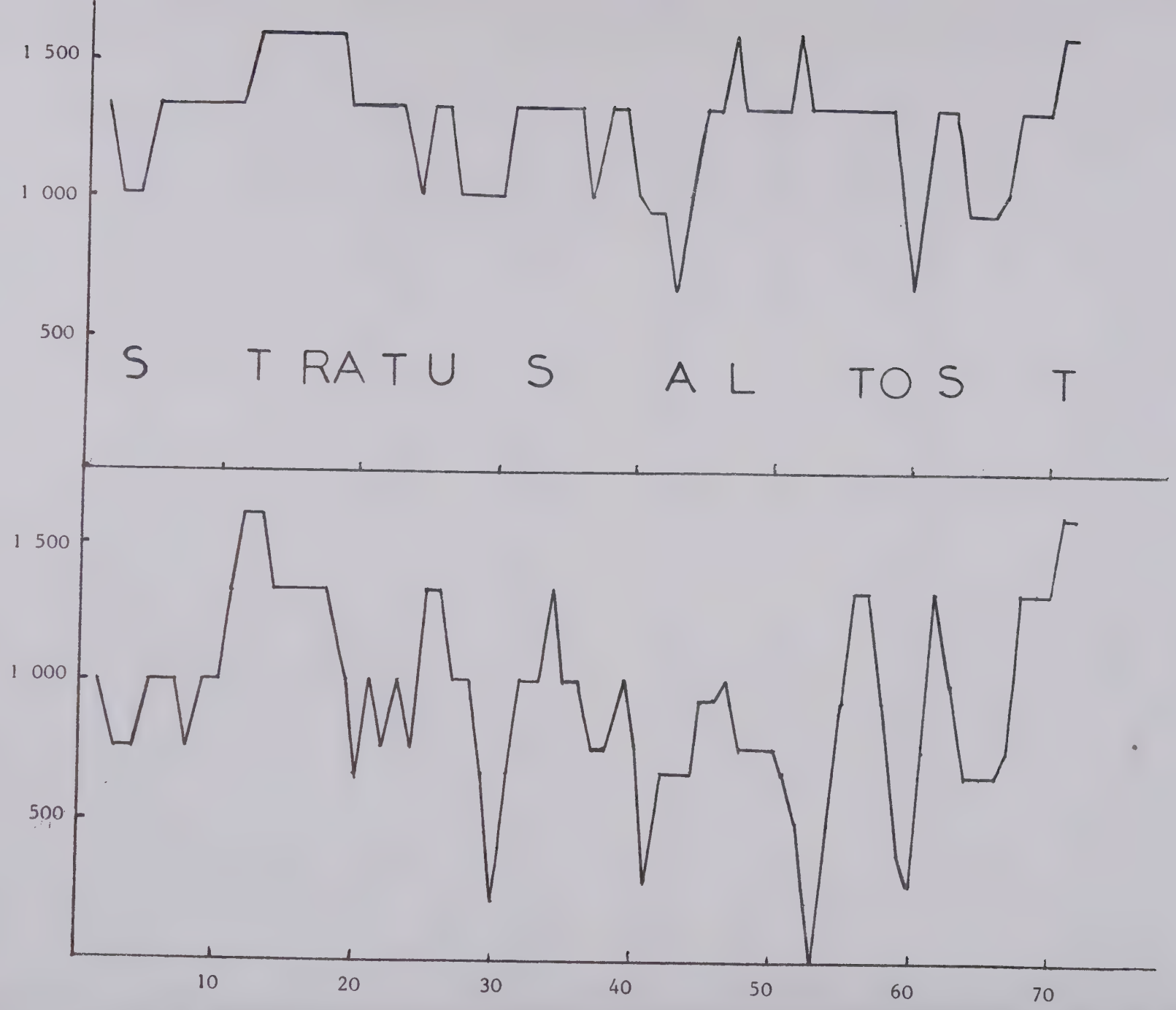
On retrouvera ce fait dans le chapitre VI : à une forme correspond en général une corrélation importante entre les opérateurs O_k, $O_{k'}$ qui extraient ses caractéristiques (valeur importante de $I[A_k ; A_{k'}]$).

Remarque générale

Mettons en parallèle les opérateurs de reconnaissance de forme et les algorithmes d'apprentissage de systèmes de reconnaissance de formes. Ces derniers peuvent utiliser des opérateurs qui calculent les "probabilités", les valeurs des fonctions H et I,.... .

Nous venons de voir que l'on pouvait trouver des corrélations entre les résultats de ces opérateurs et l'existence de formes.

On peut alors considérer la recherche en reconnaissance de forme comme un apprentissage particulier de reconnaissance de forme par cette recherche de corrélation.



Variations de $H_5[A_6]$

Variations de $I_5[A_6 ; A_7]$

III.8. EXEMPLES DE MESURE DE L'INFORMATION UTILE

Les exemples que nous allons montrer sont des mesures sur les capteurs eux-mêmes : les chapitres suivants montrerons d'autres mesures sur des opérateurs de plus haut niveau.

III.8.1. Capteur Optique

Le capteur que nous utilisons est le Calife du Laboratoire Central de l'Armement. C'est un capteur optique à balayage (flying spot) qui donne une représentation matricielle de la luminosité d'une photo (figure 22-a).

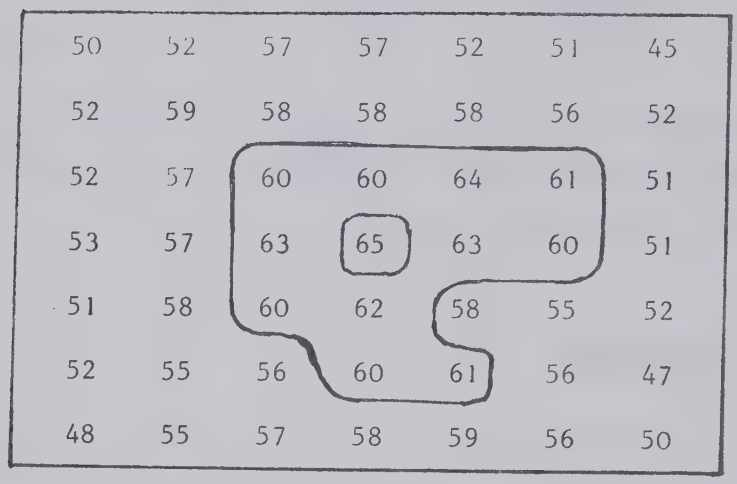
Un certain nombre d'objets blancs sur fond noir ont été photographiés puis digitalisés par le Calife.

Les formes élémentaires Ω à extraire sont {blanc, noir, bord d'objet}. 300 carrés de dimension 7 x 7 ont été découpés et présentés au "professeur". Celui-ci décide alors sur chacun des 300 carrés (ensemble d'apprentissage) laquelle des 3 formes de Ω il représente. La figure 10 montre un sous ensemble de l'ensemble d'apprentissage.

Chaque carré est un ensemble de 7 x 7 = 49 résultats d'opérateurs élémentaires O_j :

$$\{O_j \mid 1 \leqslant j \leqslant 49\}$$

En utilisant la matrice $\|p[\omega(i) \mid a_j(k)]\|$ obtenue sur l'ensemble d'apprentissage, on évalue l'information utile des opérateurs $O_j = I[\Omega ; A_j]$.



Ces quantités sont à comparer à $H[\Omega] = 143$. Un traitement sur les données A_j est donc nécessaire puisque l'information du meilleur opérateur ici l'opérateur central, est insuffisante. Voir le chapitre V.

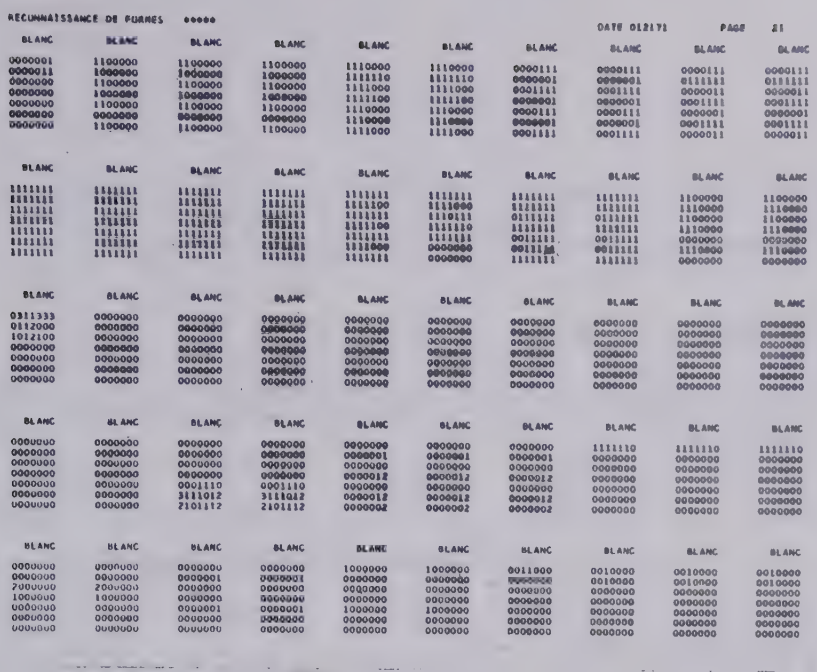
III.8.2. Capteur acoustique (corrélation inter canaux)

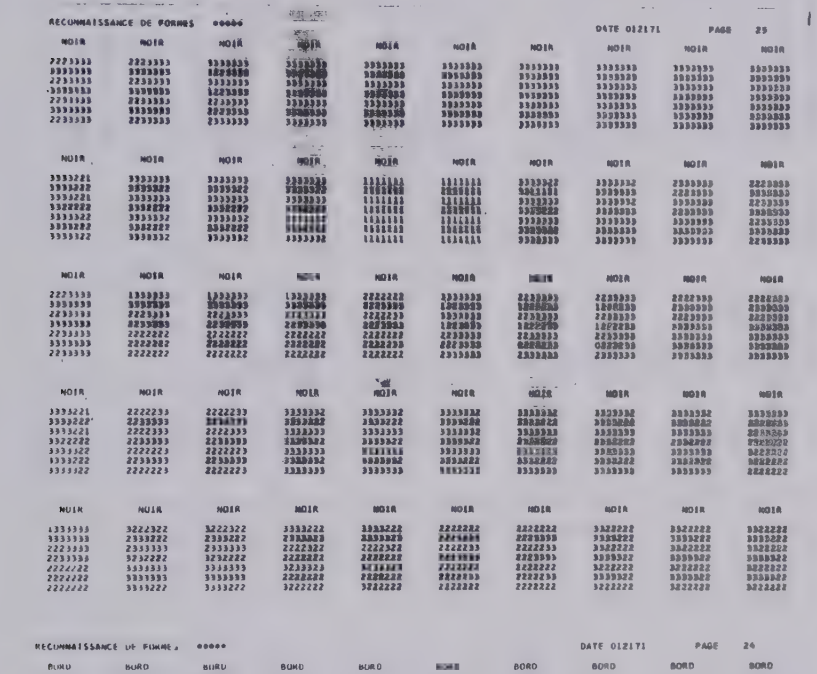
Le capteur utilisé est le vocoder à 12 canaux du Centre National d'Etudes des Télécommunications. Une trentaine de mots ont été prononcés devant le vocoder. La figure 11 donne une représentation des données fournies toutes les 23 ms par le vocoder à l'ordinateur.

Les opérateur élémentaires sont ici les canaux numérotés de 1 à 12 :

$$\{O_j \mid j \in 1, \ldots, 12\}$$

Si on représente sur un tableau les valeurs de $I[O_i ; O_j]$ pour tous les couples de canaux :





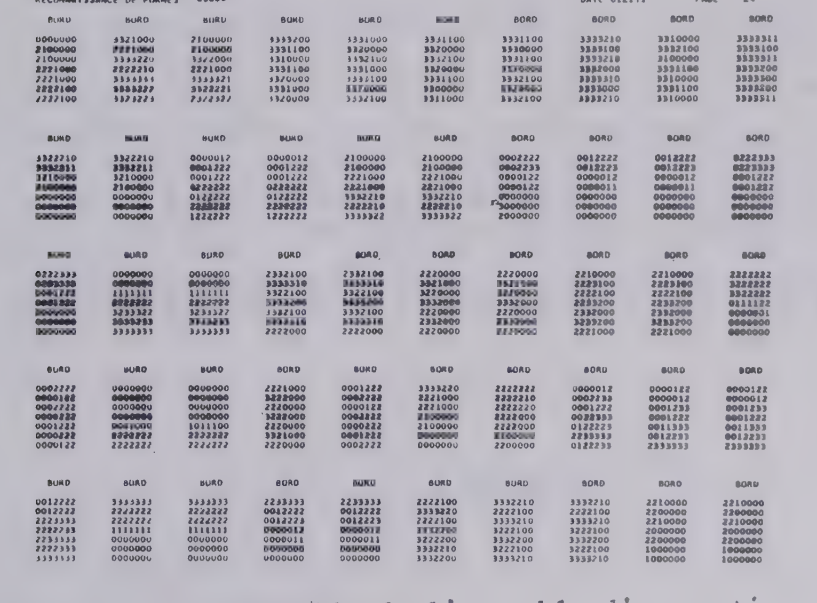


Figure 10 : Sous ensemble de l'ensemble d'apprentissage du §III.8.1.

FORTE PLUIE

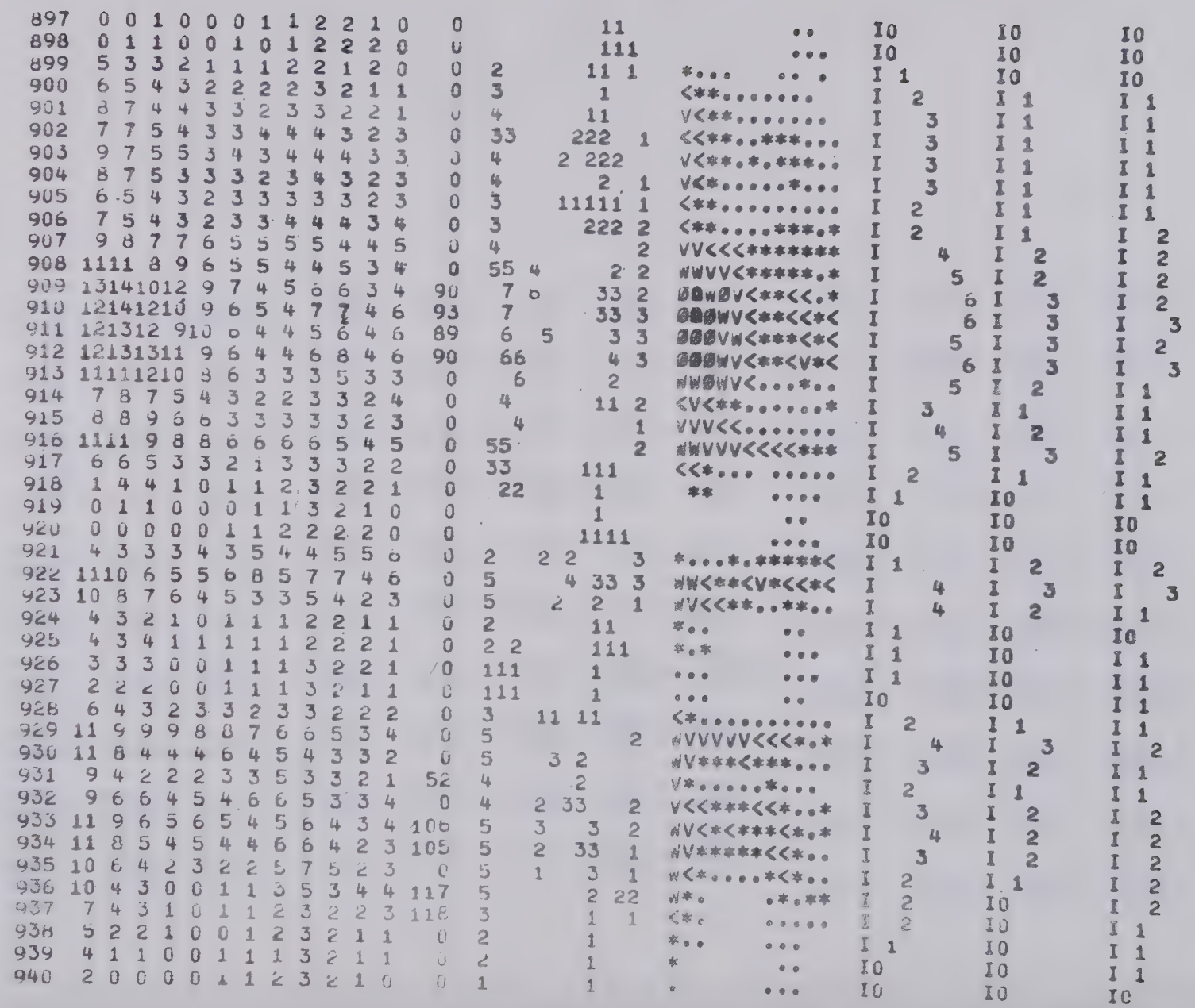


Figure 11

A	Numéro de la ligne		
B	Sorties du banc de filtres		
C	Valeurs du pitch		
D	Amplitude et position des "formants"		
E	Visualisation des sorties du banc de filtres		
F_i	Moyenne des sorties des filtres	i = 1	n° 1 au n° 4
		i = 2	n° 5 au n° 8
		i = 3	n° 9 au n° 12

Sorties du programme "VISU", lorsque les mots "forte pluie" ont été prononcés

Figure 11 bis - Explication de la figure 11

O_i / O_j	1	2	3	4	5	6	7	8	9	10	11	12
1	2178	543	326	294	273	223	254	197	228	167	162	217
2	543	2430	690	566	494	396	330	307	238	268	249	238
3	326	690	2386	922	689	605	401	286	199	260	194	202
4	294	566	922	2385	931	588	368	284	211	222	216	182
5	273	494	689	931	2273	780	403	279	200	113	214	199
6	225	396	605	588	780	2139	533	238	181	230	248	189
7	254	330	401	368	405	533	2074	466	271	316	269	250
8	197	307	286	284	279	238	466	2028	538	311	269	237
9	228	238	199	211	200	181	271	538	1890	417	254	215
10	167	268	260	222	163	230	316	311	417	1921	424	314
11	162	249	194	216	214	248	269	269	254	424	1979	450
12	217	238	202	182	199	189	250	237	215	314	450	1885

On voit sur ce tableau que les canaux n° 3, 4, 5 et 6 sont mutuellement correlés assez fortement, ainsi que les canaux 8 et 9.

III.8.3. Capteur acoustique (reconnaissance de phonème)

Supposons que nous découpions ces mêmes listes de données en segment de temps correspondant à 4 digitalisations successives. Nous disposons donc de 4 x 12 = 48 valeurs de canaux. Ces valeurs sont peut-être représentatives du son qui est émis. Considérons ces 48 valeurs comme résultats de 48 opérateurs canaux.

Nous avons mesuré leur information utile

$$I[O_j ; \Omega]$$

dans le but de reconnaître les phonèmes Ω.

24 phonèmes ont été considérés :

A, E, I, O, U, EE, IN, OU, UI, JJ, B, D, F, J, K, L, M, N, O, P, R, S, T, U

(EE : son é comme étable ; JJ : comme dans mouillé).

Leur information utile est donné par le tableau suivant :

1	2	3	4	5	6	7	8	9	10	11	12	
82	80	76	76	73	63	67	72	58	58	43	54	T - 1
92	98	96	92	84	73	76	75	70	69	54	56	T
75	81	85	88	82	73	67	75	63	53	48	48	T + 1
68	70	75	68	60	54	58	63	50	44	42	38	T + 2

On voit sur le tableau que l'information utile apportée par ce vocoder se situe principalement dans les graves : canaux 1 à 5 mais aussi vers le canal 8 (zone du 2ème formant vers 2 000 Hz).

Ceci est en accord avec le fait que les fréquences centrales des filtres du vocoder utilisé sont en progression géométrique donc que les fréquences basses sont analysées avec plus de finesse.

C'est en accord aussi avec l'expérience des phonéticiens pour qui intuitivement l'information utile se trouve principalement dans la zone des 2 formants en particulier vers 2 000 Hz.

On aurait donc intérêt sans doute à resserrer les fréquences des filtres du vocoder vers 2 000 Hz.

Le tableau de la page précédente donne des corrélations importantes entre les résultats des filtres situés à l'intérieur de ces deux zones : ce fait est conforme à l'idée émise en III.7., selon laquelle l'existence d'une forme (en particulier ici les formes caractéristiques des phonèmes), signifie que les opérateurs qui extraient les formes caractéristiques de cette forme, ont des résultats corrélés.

III.8.4. Capteur acoustique (segmentation)

En vue de résoudre le problème de la reconnaissance de la parole, on peut chercher à reconnaître dans les données vocales digitalisées en fonction du temps, les instants caractéristiques des phonèmes.

Il s'agit de partager le temps en segments ΔT correspondant le plus possible à l'émission de diphonèmes (passage d'un phonème au phonème suivant : thèse de LIENARD [43]) ; ensuite par un procédé de corrélation simple, analyser chaque segment pour reconnaître le diphonème qui a été émis pendant ce temps ΔT.

Chaque segment est borné par des "tops" de segmentation. Le problème est de déterminer ces "tops" par un opérateur travaillant sur les données vocales.

Un "professeur" indique sur des données d'apprentissage les positions estimées des "tops" (figure 12).

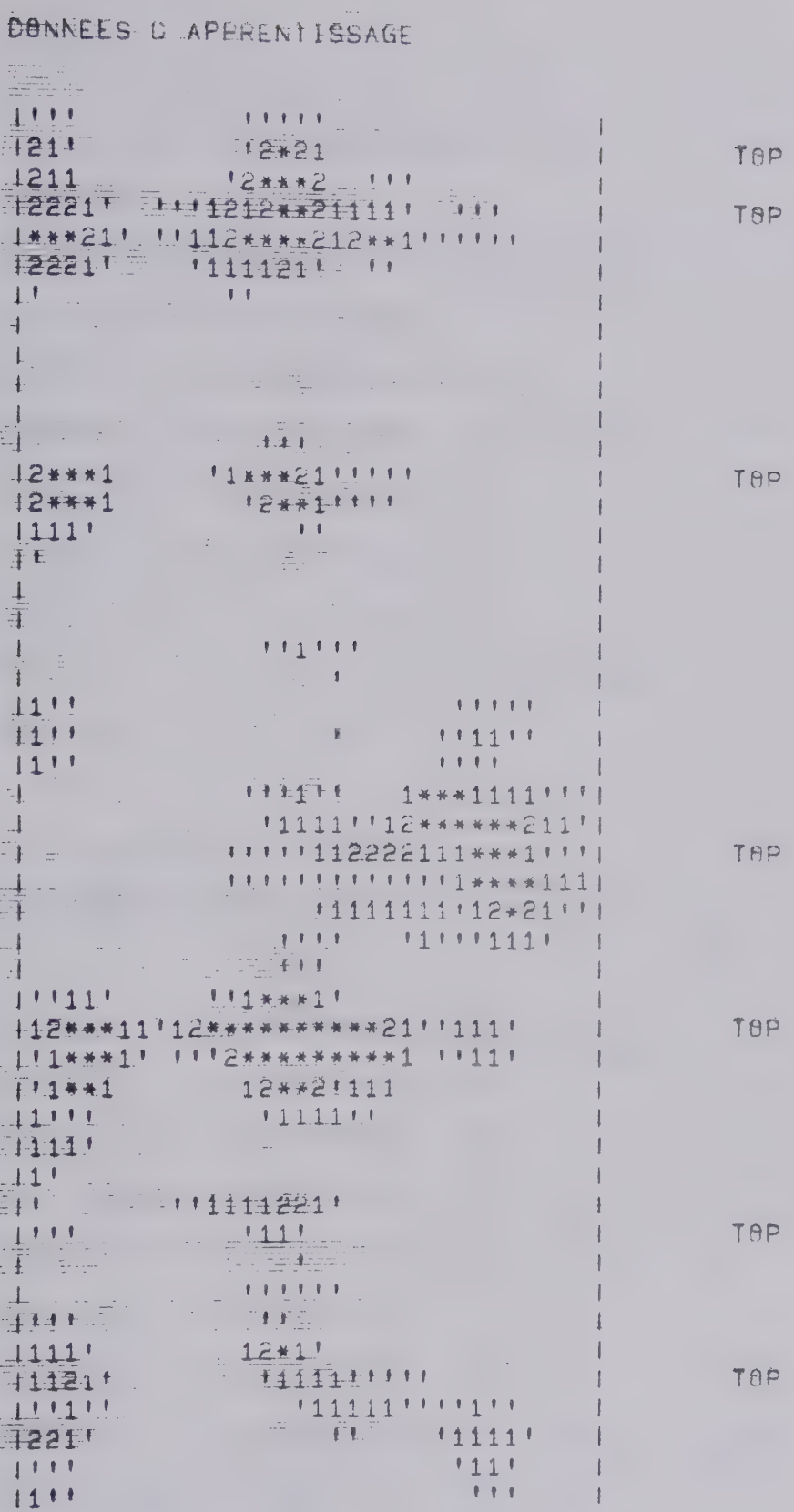


Figure 12 : Données vocales
et Tops de segmentation

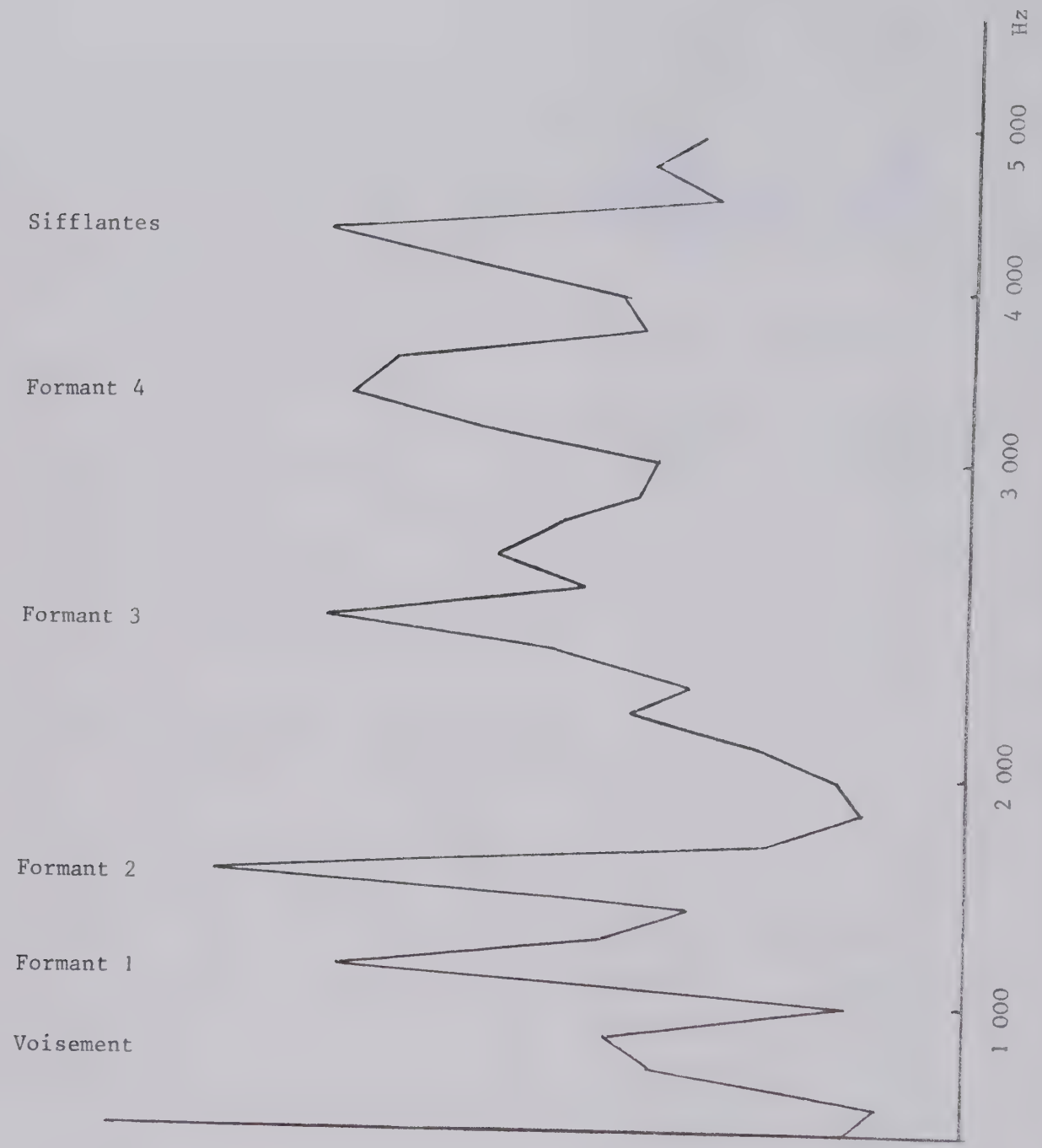


Figure 13 : valeurs de l'information utile
en fonction de la fréquence (segmentation)

Le capteur acoustique utilisé ici est un banc de 32 filtres échantillonnés toutes les 10 ms. Ce capteur est utilisé par les acousticiens (MALAVAR, LIENARD) dans leurs recherches en reconnaissance de la parole continue [43].

Nous avons mesuré l'information utile des 32 opérateurs O_j correspondant aux 32 filtres du capteur ; la figure 13 donne en fonction de j l'évolution de l'information : $I[\Omega ; A_j]$.

Les fréquences basses correspondent aux valeurs faibles de l'indice i.

La courbe présentée fournit par ses maximums les 6 plages de fréquence sur lesquelles il est intéressant de faire travailler les opérateurs de segmentation. Ces plages correspondent aux fréquences de voisement, des quatre formants et des sifflantes : fréquences (800, 1 200, 1 400, 2 400, 3 300, 4 100).

Ce résultat est conforme aux idées intuitives des phonéticiens.

FREQUENCE	INFORMATION UTILE	FREQUENCE	INFORMATION UTILE
1	0,0036	17	197
2	23	18	118
3	95	19	146
4	109	20	128
5	47	21	101
6	193	22	97
7	112	23	150
8	84	24	192
9	234	25	179
10	63	26	101
11	31	27	109
12	39	28	93
13	64	29	200
14	104	30	78
15	85	31	100
16	129	32	84

III.9. Validité de l'utilisation de l'information utile

L'information utile ainsi définie, tout le long de ce chapitre, est une fonction d'évaluation des opérateurs. Nous allons l'utiliser comme telle dans des algorithmes d'apprentissage en reconnaissance des formes (chapitres suivants).

Nous avons pris cette fonction :

1. parce qu'elle a une signification précise dans la théorie des questionnaires : elle permet l'optimisation pas à pas des questionnaires (chapitre suivant).

2. parce qu'elle satisfait un certain nombre de relations, comme nous venons de le voir : ces relations sont celles qu'on demanderait a priori pour une fonction mesurant l'"information utile".

3. parce que dans tous les problèmes d'applications où l'homme a une certaine idée des informations utiles transmises, les valeurs de la fonction $I[\Omega ; A]$ correspondent à ces quantités intuitives.

Cette fonction est par ailleurs d'un emploi très simple et peut-être calculée rapidement en ordinateur : en écrivant par avance en mémoire (ou en cablant) les premières valeurs de la fonction x Log x, on se dispense du calcul de ces valeurs et le temps mis pour calculer $I[A ; \Omega]$ est le même que celui du calcul de la probabilité d'erreur de l'algorithme.

CHAPITRE IV

UTILISATION DE L'INFORMATION MUTUELLE DANS DES ALGORITHMES SEQUENTIELS DE RECONNAISSANCE DE FORMES

IV.1. LES PSEUDOQUESTIONNAIRES

A la suite de PICARD, TERRENOIRE [88] introduit le formalisme des pseudoquestionnaires. Cette théorie étudie la manière d'optimiser l'ordre des questions séquentielles O_j à poser pour obtenir

$$\{H[\Omega \mid a_j(k), \ldots] \geqslant \epsilon\} \text{ ou bien}$$

$$\{\exists_i \mid p(\omega_i) \geqslant 1 - \epsilon'\} \text{ étant données les précisions } \epsilon \text{ et } \epsilon'$$

Dans le cas où les questions O_j sont indépendantes, les théorèmes ont été démontrés par TERRENOIRE sur la convergence vers $H[\Omega]$ de $I[\Omega ; A_{j_1}, A_{j_2}, \ldots, A_{j_n}]$: si n tend vers l'infini (pseudoquestionnaire infini) et si le choix de la question O_j à poser à chaque fois parmi les questions restantes est celle qui correspond à une information supérieure à

$$(1 - \rho) \text{ Sup } (I[\Omega ; O_j])$$

alors $I[\Omega ; A_{j_1}, A_{j_2}, \ldots, A_{j_n}]$ tend vers $H[\Omega]$.

Si on veut poser un nombre limité de questions données a priori, une méthode consiste donc à prendre à chaque itération la question correspondant au plus grand $I[\Omega ; I_j]$. Le pseudoquestionnaire obtenu n' est pas

toujours optimal, mais donne une approximation du pseudoquestionnaire optimal ($H[\Omega \mid A_{j_2}, \ldots, A_{j_n}]$ le plus petit possible pour un nombre donné de questions).

SLAGLE et LEE [86] donnent une technique pour donner une meilleure approximation du pseudoquestionnaire optimal. Elle fait appel à une recherche à travers un sous-ensemble des pseudoquestionnaires possibles. Ce parcours ressemble à la recherche du meilleur cours en théorie des jeux, avec un calcul de minimaux. Ici, le calcul du minimax est transformé en un calcul de "maximoyen", SLAGLE considère en effet la recherche du questionnaire comme un "jeu contre la nature". La fonction d'évaluation du questionnaire, est l'information mutuelle $I[A_j ; \Omega]$. La recherche se fait suivant une certaine profondeur p, le choix de la question se fait en prenant celle qui maximise l'espérance mathématique de l'information utile apportée par le pseudoquestionnaire à la profondeur p.

Dans les exemples qui vont suivre, nous avons pris la méthode simple du choix pas à pas de la question la plus importante.

Le meilleur opérateur est pris suivant la fonction :

$$(19) \qquad I[A_j ; \Omega] = \sum_{i,\,k} p(\omega(i), a_j(k)) \cdot \mathrm{Log} \frac{p(\omega(i), a_j(k))}{p(\omega(i)) \cdot p(a_j(k))}$$

A la réponse $a_j(k)$, on calcule les probabilités conditionnelles des formes $\omega(i)$ par la formule de Bayes :

$$(20) \qquad p[\omega(i) \mid a_j(k)] = p[\omega(i)] \cdot \frac{p[a_j(k) \mid \omega(i)]}{\sum_i p[\omega(i)] \cdot p[a_j(k) \mid \omega(i)]}$$

Alors les probabilités : $p[\omega(i), a_{j'}(k')] = p[\omega(i)] \cdot p[a_{j'}(k') \mid \omega(i)]$ se transforment de la même façon en ;

$$p[\omega(i) \mid a_j(k)] \cdot p[a_{j'}(k') \mid \omega(i)]$$

On suppose en effet que $p[a_{j'}(k') \mid \omega(i), a_j(k)] = p[a_{j'}(k') \mid \omega(i)]$ (approximation n° 1).

Le meilleur opérateur suivant $(O_{j'})$ est celui qui maximise la nouvelle information utile :

$$(21) \qquad I[A_{j'}; \Omega] = \sum_{i,k} p[\omega(i) \mid a_j(k)] \cdot p[a_{j'}(k') \mid \omega(i)] \cdot \operatorname{Log} \frac{p[a_{j'}(k') \mid \omega(i)]}{p[a_{j'}(k')]}$$

On applique ensuite itérativement la formule suivante si le résultat $a_{j'}(k')$ est obtenu :

$$(22) \qquad p[\omega(i) \mid a_{j'}(k'), a_j(k)] = p[\omega(i) \mid a_j(k)] \cdot \frac{p[a_{j'}(k') \mid \omega(i)]}{\sum_i p[\omega(i) \mid a_j(k)] \cdot p[a_{j'}(k' \mid \omega_i)]}$$

La formule de Bayes aurait donné :

$$(23) \qquad p[\omega(i) \mid a_{j'}(k'), a_j(k)] = p[\omega(i)] \cdot \frac{p[a_j(k), a_{j'}(k') \mid \omega(i)]}{\sum_i p[\omega(i)] \cdot p[a_j(k), a_{j'}(k') \mid \omega(i)]}$$

La formule (22) utilisée n'est exacte pour tout les $\omega(i)$ que si

$$\forall_i : p[a_j(k), a_{j'}(k') \mid \omega(i)] = p[a_j(k) \mid \omega(i)] \cdot p[a_{j'}(k') \mid \omega(i)]$$

c'est-à-dire si $a_j(k)$ et $a_{j'}(k')$ sont indépendantes. Si elles ne sont pas indépendantes, (cas général) la formule (22) ne donne qu'une approximation de $p[\omega(i) \mid a_{j'}(k'), a_j(k)]$. Mais la formule (22) conduit à un calcul beaucoup plus simple et surtout à une occupation de la mémoire de l'ordinateur beaucoup plus faible : le tableau des probabilités $p[\omega(i), a_j(k)]$ suffit en effet. C'est l'approximation n° 2.

On itère ensuite le procédé en appliquant les formules (21) et (22) jusqu'à ce que l'une des

$$p[\omega(i) \mid a_{j_1}(k_1), a_{j_2}(k_2), \ldots, a_{j_n}(k_n)]$$

dépasse un certain seuil donné, ou que l'on ait utilisé un nombre donné d'opérateurs.

IV.2. RECONNAISSANCE DE MOTS

Au chapitre I nous avons montré comment un système de reconnaissance peut se décomposer en plusieurs niveaux d'opérateurs correspondant à différents niveaux de formes. En particulier, pour la lecture de mots les différents étages à considérer sont :

résultats du capteur → résultats du prétraitement → primitives de lettres → lettres → mots

Considérons ici seulement l'étage (lettres → mots). Dans l'expérience présentée les mots à reconnaitre comprennent 6 lettres au plus, comme

D	U	P	O	N	T
1	2	3	4	5	6

Chaque lettre est repérée par sa position de 1 à 6.

Les opérateurs de reconnaissance que nous avons utilisés sont les :

$$\{O_j = \text{quelle est la lettre placée en } j \ ? \mid j \in 1, \ldots, 6\}$$

On remarque que chacun de ces opérateurs est l'opérateur de reconnaissance de lettres factorisé et muni du paramètre j indiquant la position du sous ensemble de données sur lequel le système de reconnaissance de lettres va travailler.

Dans un premier temps, nous avons simulé le programme de reconnaissance de lettres et supposé qu'il donne comme résultat un vecteur de probabilité $p[\omega_1(i)]$ sur les lettres : la lettre reconnue n'est reconnue qu'avec une probabilité $p[\omega_1(i)] = 0,8$.

Nous avons choisi empiriquement cent mots courants du vocabulaire, de manière à constituer un bon échantillonnage du vocabulaire français. A chacun de ces mots nous avons affecté une probabilité a priori $p[\omega_m(i)]$

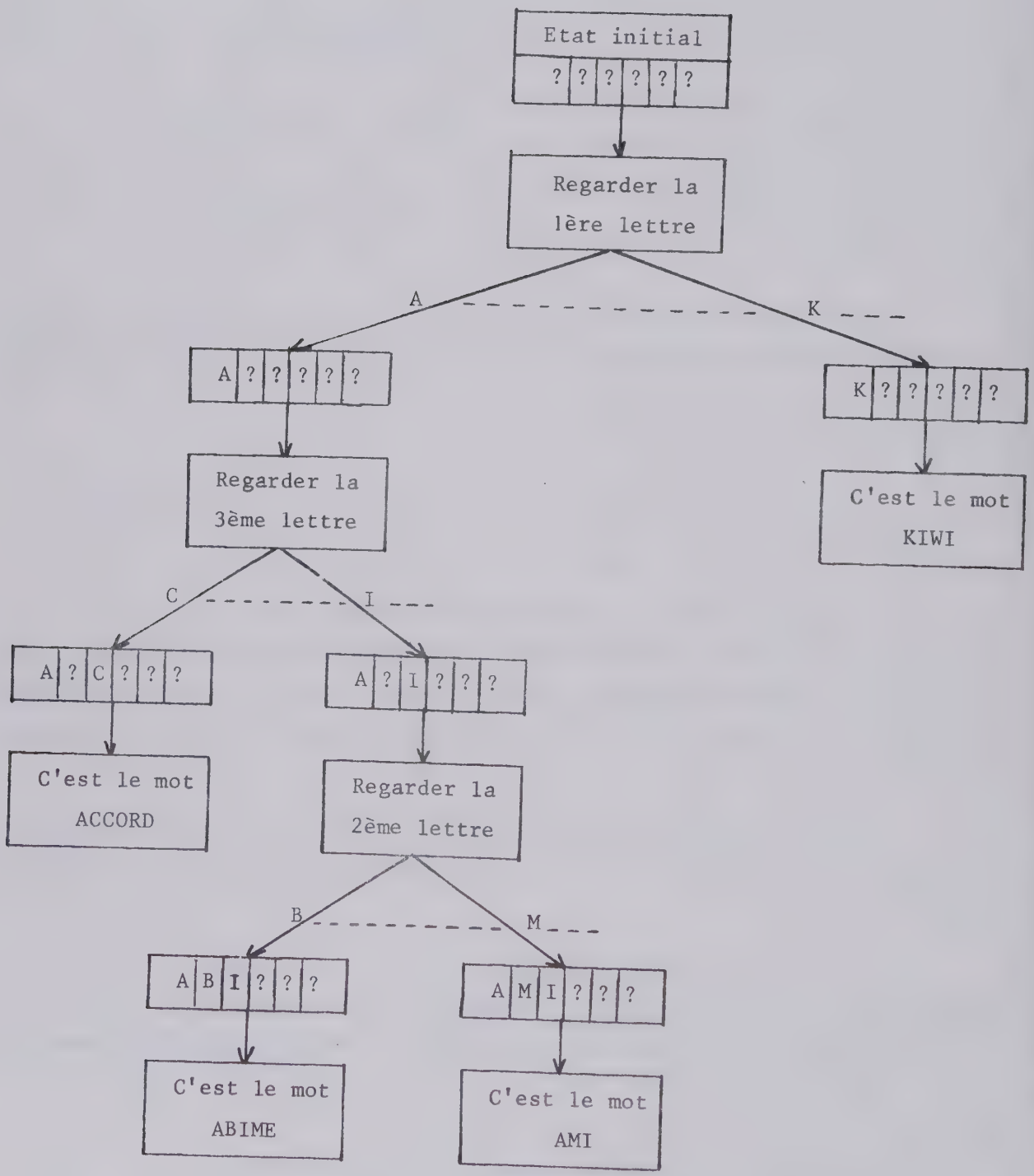
La méthode décrite au paragraphe précédent a été utilisée. En prenant un seuil de probabilité de 70 % sur les mots pour prendre une décision, on obtient un nombre moyen de lectures de lettres de 2. En faisant un choix aléatoire de l'opérateur O_j et en utilisant la formule de Bayes, ce nombre moyen atteint 2,7.

Par cette méthode, on établit en 2 mn (CDC 3600) un arbre de décision que l'on représente en ordinateur sous forme de liste : voir ROCHE et HEUDE [66].

Cet arbre est implanté au fur et à mesure des besoins. Si le programme doit lire un mot déjà connu, l'algorithme de reconnaissance fonctionne suivant l'arbre de décision déjà en mémoire. Si le programme lit un mot nouveau, l'algorithme s'enrichit et l'arbre de décision acquiert une nouvelle branche.

Une fois l'implantation achevée, la lecture est très rapide (qqs 100 μs) et l'occupation mémoire est de 1 K mots.

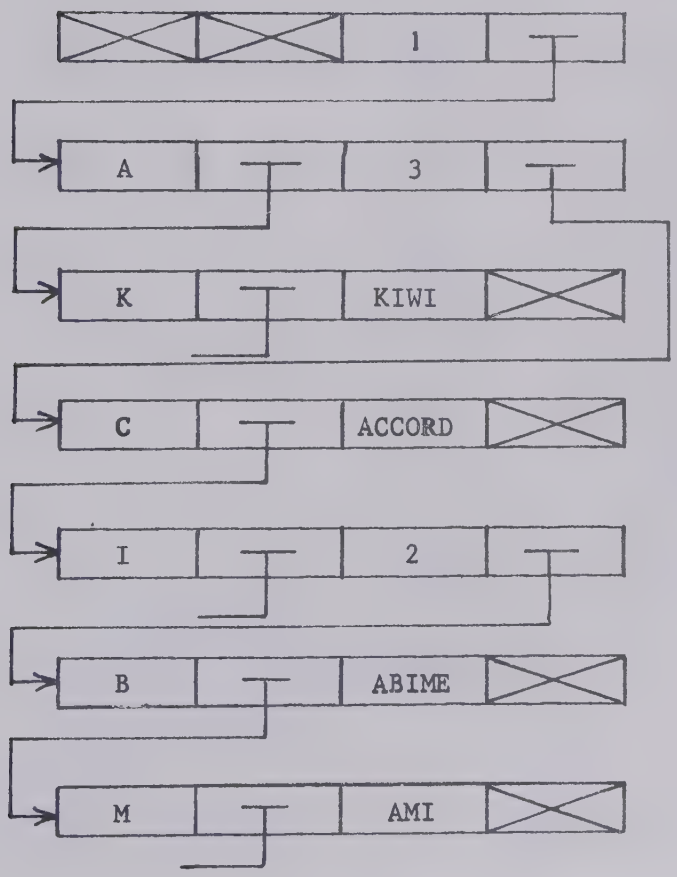
Prenons une partie de l'arbre de décision obtenu (voir §I.2.1 et [66]).



Cette partie de l'arbre de décision est implantée au fur et à mesure des apparitions des mots, sous forme de liste de cellules. Quatre cellules indiquent à chaque décision à prendre : elles contiennent :

- la réponse à la dernière question posée
- un pointeur vers la réponse suivante
- la décision à prendre : numéro de la question à poser, ou forme reconnue
- un pointeur vers la première des réponses possibles à la question à poser.

Dans le cas considéré :



BREAK	JOUER	SANS
CADRAN	ZOUAVE	MOI
CAS	QUARTZ	MIETTE
CERISE	QUOI	CYCLE
LANGUE	ACQUIS	DAHLIA
LE	OMBRE	DECIDE
ANNEXE	PLAQUE	MOITIE
ARGOT	NEGOCE	MUGUET
JOUET	NATION	OUVRIR
LINGOT	EPIQUE	PAGODE
LOCK	POSE	PARURE
LAIQUE	RAILLA	PAILLE
SIEGE	RANCON	PENAUD
NICHE	REFUGE	PHOBIE
BICHE	ETANG	BIELLE
NORMAL	FONCER	BLANC
SOUCHE	FRUIT	PIEGE
DUPONT	SPORT	AMI
DESTIN	TENEUR	
DIRE	TIMIDE	
OASIS	ABRUTI	
TUILE	MAIGRE	
UN	MAILLE	
DOUANE	MEGERE	
VALISE	MYOPE	
VIOLON	LUXE	
DRU	ACCORD	
ECLAIR	POUPEE	
EFFET	FECOND	
GAIETE	TRIO	
GENTIL	BACHOT	
GROS	BAYER	
HACHE	AZUR	
IDIOT	CHAMP	
IGNARE	COMITE	
INDEX	CROIX	
IRONIE	PRESTE	
FIBRE	RELIEF	
RACINE	KIWI	
MUSCLE	RIDEAU	
ABIME	ROSEE	

Figure 14-a : Vocabulaire utilisé

```
LA FORME A LIRE EST :PENAUD


MOT NOUVEAU,L ARBRE NE POSSEDE RIEN
ON AJOUTE LECTURE EN :1
ON AJOUTE LECTURE EN :4
ON AJOUTE LECTURE EN :5
NOUS AVONS LU :
PENAUD


LA FORME A LIRE EST :JOUER


MOT NOUVEAU,L ARBRE POSSEDE DEJA:
LECTURE EN:1
ON AJOUTE LECTURE EN :5
NOUS AVONS LU :
JOUER


LA FORME A LIRE EST :BREAK


MOT NOUVEAU,L ARBRE POSSEDE DEJA:
LECTURE EN:1
ON AJOUTE LECTURE EN :5
NOUS AVONS LU :
BREAK


LA FORME A LIRE EST :QUARTZ


MOT NOUVEAU,L ARBRE POSSEDE DEJA:
LECTURE EN:1
ON AJOUTE LECTURE EN :3
NOUS AVONS LU :
QUARTZ


LA FORME A LIRE EST :MEGERE


MOT NOUVEAU,L ARBRE POSSEDE DEJA:
LECTURE EN:1
ON AJOUTE LECTURE EN :4
ON AJOUTE LECTURE EN :3
NOUS AVONS LU :
MEGERE
```

Figure 14-b

Implantation du questionnaire

(début)

```
LA FORME A LIRE EST :POUPEE

NOUS AVONS LU :
POUPEE

LA FORME A LIRE EST :QUARTZ

NOUS AVONS LU :
QUARTZ

LA FORME A LIRE EST :ZOUAVE

NOUS AVONS LU :
ZOUAVE

LA FORME A LIRE EST :RAILLA

MOT NOUVEAU,L ARBRE POSSEDE DEJA:
LECTURE EN:1
LECTURE EN:3
ON AJOUTE LECTURE EN :5
NOUS AVONS LU :
RAILLA

LA FORME A LIRE EST :NEGOCE

NOUS AVONS LU :
NEGOCE

LA FORME A LIRE EST :NICHE

NOUS AVONS LU :
NICHE
```

Figure 14-c : Implantation du questionnaire (après 40 mots)

Dans une deuxième expérience que nous avons faite, les probabilités des mots ne sont pas déterminées a priori mais calculées en cours d'implantation de l'arbre de décision.

Les probabilités sont toutes égales au départ. Deux vecteurs de probabilités sont utilisés alternativement. L'implantation de l'arbre se fait en tenant compte de l'un ; l'autre est déterminé au fur et à mesure que l'arbre est utilisé. Périodiquement, on compare les deux vecteurs. Si la différence est trop grande, on annule le vecteur le plus ancien et on reconstruit l'arbre sur le vecteur le plus récent en utilisant le premier pour recalculer les probabilités. Si la différence n'est pas trop grande, on ne recalcule pas l'arbre complètement mais on réordonne ses branches par probabilités décroissantes afin de minimiser la longueur moyenne de l'exploration.

Les résultats ont été intéressants et en accord avec les expériences physiologiques en lecture rapide : deux lettres suffisent en moyenne pour déterminer un mot avec une bonne probabilité. La lettre la plus informante est la 1ère ; la suivante, la 3ème en général. Si la 1ère lettre est une voyelle, la suivante est soit la 2ème soit la 4ème.

Les pages précédentes montrent le vocabulaire utilisé et l'implantation de l'arbre.

IV.3. RECONNAISSANCE DE LETTRES

Un programme d'analyse de lettres a été écrit dans le but de lire des mots automatiquement. Le paragraphe I.2.1. décrit l'algorithme utilisé pour passer d'une représentation par points sur une rétine, à une représentation par un tableau des relations des primitives de la lettre. La figure 2 montre un exemple de traitement de la lettre A.

Sur le tableau des relations entre les primitives de la lettre, on veut trouver un algorithme qui reconnaît la lettre en utilisant la méthode séquentielle exposée précédemment. Voir [13].

Les opérateurs O'_j que nous avons utilisés sont les neuf suivantes :

- nombre de primitives de la lettre
- nombre de relations du tableau
- nombre de primitives n'intervenant que dans une relation
- dimension de la base des cycles élémentaires du graphe de la lettre
- nombre de primitives liées à d'autres par leur milieu
- nature de la première relation
- valeur de l'angle de la première relation
- valeur de l'angle de la deuxième relation
- valeur de l'angle de la troisième relation.

L'expérience montre que le programme n'appelle pratiquement que les opérateurs n° 1, 2 et 8 et que ceux-ci sont suffisants pour une lecture à probabilité 0,8.

Le programme a servi d'entrée à un programme de reconnaissance de mots utilisant la méthode du paragraphe précédent ;

Ici au lieu de chercher une probabilité de 0,8 pour la lettre lue avant de passer à la lettre la plus informante suivante, on modifie les probabilités pas à pas à la fois des mots et des lettres au fur et à mesure que s'effectuent les opérateurs O'_j.

On décide au niveau du mot quelle lettre lire, ensuite quel opérateur O'_j. Suivant le résultat les probabilités des mots changent et on itère le procédé jusqu'à l'obtention d'une probabilité 0,9 sur un mot.

L'algorithme converge assez vite. Avec un dictionnaire de 50 mots et une longueur de mot de 5 lettres, le nombre moyen de questions posées est de quatre seulement.

Quelques exemples : (le vocabulaire utilisé ne formait pas un échantillonnage très représentatif de la langue française, ce qui explique que la lettre la plus informante soit ici la 2ème contrairement aux expériences du chapitre précédent). Voir [13].

Reconnaissance du mot KYSTE

Lettre n° 2 ; opérateur n° 2 ; nouvelles probabilités :

ETRE 0,53
BLEU 0,37
BLANC 0,06
KYSTE 0,04

Lettre n° 3 ; opérateur n° 2 ; nouvelles probabilités :

BLEU 0,87
BLANC 0,10
KYSTE 0,03

Lettre n° 3 ; opérateur n° 9 ; nouvelles probabilités :

BLEU 0,79
BLANC 0,13
KYSTE 0,06

Lettre n° 3 ; opérateur n° 3 ; nouvelles probabilités :

KYSTE 1,00

Reconnaissance du mot ZEBRE

Lettre n° 2 ; opérateur n° 2 :

SEINE 0,29
JAUNE 0,25
CANNE 0,11
MAIS 0,07
FAIT 0,06

ZEBRE 0,004

Lettre n° 1 ; opérateur n° 2 :

JAUNE 0,48
FAIT 0,22
SEINE 0,19
JEUDI 0,05
SAIDA 0,02
ZEBRE 0,016

Lettre n° 1 ; opérateur n° 1 :

JAUNE 0,48
FAIT 0,22
SEINE 0,19
JEUDI 0,05
ZEBRE 0,03

Lettre n° 1 ; opérateur n° 8 :

ZEBRE 1,00

Reconnaissance du mot HELAS

Lettre n° 2 ; opérateur n° 2 :

SEINE 0,29
JAUNE 0,25
CANNE 0,11
MAIS 0,07
FAIT 0,06

HELAS 0,0006

Lettre n° 1 ; opérateur n° 2 :

JAUNE 0,48
FAIT 0,22
SEINE 0,19
JEUDI 0,05
SAIDA 0,022
ZEBRE 0,016
YEUX 0,013
HELAS 0,002

Lettre n° 1 ; opérateur n° 1 :

JAUNE	0,44	ZEBRE	0,03
FAIT	0,41	SAIDA	0,007
SEINE	0,06	YEUX	0,006
JEUDI	0,05	HELAS	0,004

Lettre n° 1 ; opérateur n° 8 :

HELAS 1,00

Reconnaissance du mot SAIDA

Lettre n° 2 ; opérateur n° 2 :

SEINE	0,29	EAUX	0,05
JAUNE	0,24	TARN	0,04
CANNE	0,11	RANG	0,038
MAIS	0,07	SAIDA	0,034
FAIT	0,06		

Lettre n° 1 ; opérateur n° 2 :

SEINE 0,65
CANNE 0,18
RANG 0,08
SAIDA 0,07

Lettre n° 5 ; opérateur n° 8 :

SAIDA 1,00

IV.4. RECONNAISSANCE DE LA PAROLE

Nous disposions de données provenant du vocoder du CNET à 12 canaux.

C'est un ensemble de 12 filtres passe-bande dont les fréquences centrales sont échelonnées en progression géométrique. La sortie de chaque filtre est digitalisée (logarithmiquement) et fournit pour chaque filtre une valeur entière de 0 à 15.

Une treizième valeur (pitch) est fournie par le vocoder ; elle représente l'inverse de la valeur de la fréquence fondamentale du son. Si cette valeur est nulle, le son n'est pas "voisé", ce qui arrive en général lors de l'émission de consonnes.

Un ensemble de ces 13 valeurs est fourni toutes les 23 millisecondes.

Un exemple de la visualisation de ces valeurs en forme de sonogramme est donné figure 11 : les mots forte pluie avaient été prononcés.

Les données totales dont nous disposions correspondaient à la prononciation par un locuteur des mots de la liste suivante :

Altocumulus	Averses récentes de neige
Altostratus	Orages récents
Cirrostratus	Brouillard
Cirrus	Bruine
Cumulonimbus	Forte bruine
Cumulus	Pluie
Nimbostratus	Forte pluie
Stratocumulus	Pluie et neige
Stratus	Pluie et neige forte
Forte	Neige
Fumée	Forte neige
Orage	Averses
Bruine récente	Fortes averses
Averses récentes	Averses de neige

Ces données nous ont été obligeamment prêtées par le Laboratoire de la CGE à Marcoussis.

Avec le programme de visualisation et la connaissance des mots prononcés, on a pu indiquer à l'ordinateur les sons prononcés ainsi que leurs positions dans la liste des données. On dispose pratiquement de 275 échantillons servant d'ensemble d'apprentissage.

Nous avons voulu appliquer la méthode séquentielle décrite précédemment. Les opérateurs O_j choisis devaient être a priori des fonctions ne devant dépendre qu'à un facteur près de l'intensité de la voix ainsi que de sa fréquence ; les deux axes correspondants étant logarithmiques, les opérateurs ont été choisis invariants en translation suivant les deux axes

17 opérateurs ont été choisis, en particulier :

- Pitch : cet opérateur donne deux réponses : il y a un pitch, le pitch est nul.
- Max-min : il calcule la valeur du maximum d'intensité et du minimum sur une ligne et fait la différence ; plusieurs résultats sont groupés dans une même réponse. Ceci est d'ailleurs vrai pour tous les opérateurs.
- 1 Max-1 min : il calcule la différence d'intensité entre le 1er maximum et le minimum relatif qui le suit immédiatement à droite sur la ligne.
- Variations de la moyenne : il calcule la variation de la moyenne des intensités entre la ligne que l'on traite et la ligne précédente.
- Variation en moyenne : il calcule pour chaque canal la variation par rapport à la mesure précédente dans le temps de l'intensité mesurée. Cet opérateur fait ensuite la somme des valeurs absolues de ces variations pour les 12 canaux.
- Nombre de formants : un formant est un maximum relatif de l'intensité sur une ligne.
- Nombre de formants seconds : au lieu de travailler sur la ligne brute, on calcule d'abord la ligne dite "des différences secondes", c'est-à-dire une sorte de dérivée seconde de la ligne des données. Tous les

accidents de la courbe initiale (variations de la concavité) sont donc amplifiées, et l'on trouve un nombre plus élevé de formants.

Ecart entre formants : ce sont les différences d'abscisse (ou de numéros de canaux) des divers formants.

Ecarts d'intensités entre formants : le nom de ces opérateurs suffit à expliquer leur fonction.

Notons encore :

L'opérateur Moyenne non invariant par translation, mais qui fournit cependant de précieux renseignements, l'opérateur Rang du premier formant, non invariant par translation aussi et encore deux opérateurs Surface 1 et Surface 2 dont le rôle est d'apprécier la grandeur de certains pics relatifs de la courbe des intensités.

Ce programme travaillant sur les données d'apprentissage crée un fichier contenant les 17 tableaux des probabilités $p[\omega(i), a_j(k)]$ correspondant à ces 17 opérateurs.

Les premiers de ces tableaux sont indiqués figure 15.

Les formes $\omega(i)$ correspondent aux 24 phonèmes définis par les symboles suivants :

A, E, I, O, U, EE, AN, IM, OU, UI, JJ, B, D, F, J, K, L, M, N, P, R, S, T, V

(EE : comme dans <u>é</u>glise ; JJ : comme dans mouil<u>l</u>é)

<u>Remarque sur le calcul de ces tableaux</u>

Le nombre d'échantillons étant faible pour déterminer ces probabilités, il peut arriver que l'une des probabilités, approchée par la fréquence des évènements, soit nulle, alors qu'elle devrait être seulement petite. La formule de Bayes utilisée conduit alors à l'élimination d'un phonème qui en fait est peut être celui que la chaîne d'opérateur est en train de reconnaî-

NPH	A	E	I	O	U	EE	AN	IM	OU	UI	JJ	B	D	F	J	K	L	M	N	P
APR	.09	.01	.04	.09	.10	.09	.04	.02	.01	.02	.00	.03	.01	.03	.03	.02	.03	.02	.04	.02

OPERATEUR NO : 1

REP NO :	PROBCO																				PRK
	A	E	I	O	U	EE	AN	IM	OU	UI	JJ	B	D	F	J	K	L	M	N	P	
1	.075	.009	.030	.077	.080	.075	.035	.020	.011	.014	.007	.020	.009	.004	.024	.009	.022	.017	.033	.004	.671
2	.004	.004	.007	.004	.007	.004	.004	.004	.004	.007	.004	.011	.004	.028	.004	.014	.009	.004	.007	.017	.329

OPERATEUR NO : 2

REP NO :	PROBCO																				PRK
	A	E	I	O	U	EE	AN	IM	OU	UI	JJ	B	D	F	J	K	L	M	N	P	
1	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.062
2	.006	.003	.003	.003	.006	.003	.003	.003	.003	.003	.003	.003	.009	.003	.012	.004	.003	.003	.004	.166	
3	.007	.003	.003	.003	.003	.006	.006	.004	.003	.003	.003	.004	.006	.007	.006	.003	.004	.003	.003	.003	.111
4	.028	.003	.003	.003	.004	.012	.003	.004	.003	.003	.003	.007	.003	.003	.009	.003	.003	.003	.006	.004	.142
5	.011	.006	.006	.003	.011	.016	.011	.007	.004	.004	.003	.004	.003	.006	.006	.003	.003	.003	.007	.006	.145
6	.006	.003	.006	.009	.019	.014	.009	.004	.004	.006	.003	.007	.003	.003	.003	.003	.011	.004	.009	.004	.141
7	.003	.003	.007	.011	.016	.006	.004	.003	.004	.006	.004	.004	.003	.003	.003	.003	.006	.003	.006	.003	.110
8	.003	.003	.009	.034	.009	.006	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.009	.004	.003	.122

OPERATEUR NO : 3

REP NO :	PROBCO																				PRK
	A	E	I	O	U	EE	AN	IM	OU	UI	JJ	B	D	F	J	K	L	M	N	P	
1	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.062
2	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.062
3	.003	.006	.021	.021	.051	.011	.004	.007	.004	.011	.004	.012	.006	.015	.006	.011	.015	.011	.019	.007	.322
4	.003	.003	.003	.004	.004	.007	.006	.007	.003	.003	.003	.007	.003	.004	.007	.003	.004	.003	.006	.003	.112
5	.018	.003	.003	.029	.003	.034	.018	.003	.004	.003	.003	.003	.003	.003	.006	.004	.003	.003	.003	.006	.198
6	.009	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.076
7	.022	.003	.003	.003	.003	.003	.003	.003	.004	.003	.003	.003	.003	.003	.004	.003	.003	.003	.003	.003	.096
8	.006	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.071

OPERATEUR NO : 4

REP NO :	PROBCO																				PRK
	A	E	I	O	U	EE	AN	IM	OU	UI	JJ	B	D	F	J	K	L	M	N	P	
1	.007	.002	.002	.005	.002	.002	.006	.002	.002	.002	.002	.004	.002	.004	.002	.004	.002	.002	.002	.002	.076
2	.018	.002	.002	.009	.002	.004	.011	.002	.004	.002	.002	.004	.002	.002	.004	.002	.002	.002	.006	.002	.109
3	.012	.004	.002	.004	.005	.004	.004	.002	.002	.005	.002	.005	.002	.002	.002	.002	.005	.002	.004	.004	.100
4	.006	.002	.004	.002	.004	.005	.002	.005	.004	.002	.002	.002	.002	.005	.004	.002	.002	.004	.002	.004	.096
5	.004	.004	.002	.002	.002	.026	.002	.005	.002	.002	.002	.002	.004	.002	.007	.002	.005	.004	.007	.004	.118
6	.004	.002	.002	.013	.026	.005	.004	.005	.004	.002	.004	.007	.004	.009	.004	.006	.007	.006	.006	.002	.146
7	.002	.002	.015	.015	.015	.007	.002	.002	.002	.005	.002	.004	.002	.002	.002	.005	.004	.002	.004	.002	.114
8	.002	.002	.004	.005	.004	.002	.004	.002	.002	.004	.002	.002	.002	.002	.004	.002	.002	.002	.002	.002	.067
9	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.004	.058
10	.002	.002	.002	.004	.002	.002	.002	.002	.002	.002	.002	.002	.002	.004	.002	.002	.002	.002	.002	.004	.061
11	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.002	.055

OPERATEUR NO : 5

REP NO :	PROBCO																				PRK
	A	E	I	O	U	EE	AN	IM	OU	UI	JJ	B	D	F	J	K	L	M	N	P	
1	.007	.003	.018	.026	.008	.030	.005	.010	.007	.007	.003	.008	.003	.010	.007	.012	.010	.007	.007	.005	.256
2	.020	.007	.008	.012	.033	.012	.008	.003	.005	.005	.003	.003	.005	.007	.008	.003	.007	.007	.014	.005	.209
3	.020	.003	.003	.005	.020	.008	.007	.005	.003	.005	.005	.003	.003	.005	.008	.003	.007	.005	.007	.003	.173
4	.014	.003	.003	.012	.003	.005	.005	.005	.003	.005	.003	.005	.003	.003	.003	.005	.005	.003	.005	.003	.125
5	.009	.003	.003	.007	.003	.003	.010	.003	.003	.003	.003	.008	.003	.005	.003	.003	.003	.003	.003	.005	.102
6	.003	.003	.003	.008	.007	.010	.005	.003	.003	.003	.003	.007	.005	.005	.003	.003	.003	.003	.005	.007	.135

OPERATEUR NO : 6

REP NO :	PROBCO																				PRK
	A	E	I	O	U	EE	AN	IM	OU	UI	JJ	B	D	F	J	K	L	M	N	P	
1	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.006	.003	.003	.003	.003	.075
2	.020	.003	.003	.003	.008	.003	.012	.004	.003	.003	.003	.003	.003	.003	.006	.009	.004	.003	.003	.003	.173
3	.029	.003	.004	.008	.019	.029	.009	.009	.004	.004	.003	.008	.003	.015	.012	.003	.004	.003	.013	.006	.240
4	.006	.004	.017	.012	.025	.022	.004	.004	.003	.006	.004	.009	.006	.004	.004	.003	.015	.006	.004	.004	.213
5	.003	.004	.004	.009	.006	.004	.006	.003	.004	.006	.003	.006	.003	.004	.003	.003	.003	.004	.009	.006	.122
6	.003	.003	.004	.031	.008	.003	.003	.003	.004	.003	.003	.003	.003	.003	.003	.003	.003	.006	.004	.003	.108
7	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.070

OPERATEUR NO : 7

REP NO :	PROBCO																				PRK
	A	E	I	O	U	EE	AN	IM	OU	UI	JJ	B	D	F	J	K	L	M	N	P	
1	.003	.003	.005	.003	.005	.005	.003	.003	.003	.005	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.067
2	.008	.005	.022	.024	.020	.014	.005	.003	.007	.007	.005	.008	.007	.016	.014	.005	.007	.007	.012	.007	.302
3	.030	.005	.003	.016	.024	.027	.007	.010	.003	.005	.003	.005	.003	.007	.012	.014	.008	.008	.008	.003	.241
4	.022	.003	.003	.016	.020	.008	.018	.007	.005	.005	.003	.008	.003	.005	.003	.003	.005	.003	.008	.005	.173
5	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.003	.071
6	.003	.003	.003	.008	.003	.010	.005	.003	.003	.003	.003	.007	.003	.005	.003	.003	.003	.003	.005	.007	.125

Figure 15 : Reconnaissance de phonèmes

tableau des $p[a_j(k), w(i)]$

tre. Pour remédier à ce défaut nous avons ajouté systèmatiquement une valeur faible (2.10^{-3} ou 3.10^{-3}) aux probabilités, ce qui a éliminé certaines discontinuités remarquées dans l'évolution des probabilités $p[\omega(i)]$.

Un phénomène analogue se retrouve dans la détermination de certains filtres linéaires : voir JOUANNAUD [32] et [44] :

S'il n'y a aucun bruit la convergence du calcul du filtre est accélérée en introduisant artificiellement un bruit virtuel faible.

Un certain nombre de résultats obtenus sont indiqués figure 16 et 17.

Le premier de ces résultats concerne l'analyse de la prononciation du mot stratus. A chaque ligne des données correspondantes, le traitement séquentiel utilise systématiquement 10 itérations du procédé en prenant les 10 meilleurs opérateurs au sens de l'information mutuelle.

Après chaque itération, la ligne de valeurs indiquées représente les probabilités conditionnelles des formes $\omega(i)$ après l'itération correspondante.

Après chaque traitement de ligne de données nous décidons que le phonème qui est entrain d'être prononcé est celui qui correspond à la plus grande probabilité. Nous obtenons alors pour le mot STRATUS la suite suivante de phonèmes.

Λ S S S S S R Λ Λ Λ T T R A A A R R R R R T U U U U R S S S S S S S R S R S S Λ

(le symbole Λ signifie qu'un silence a été détecté et qu'aucun traitement n'a été fait : la somme des valeurs sur une ligne est inférieure à un certain seuil).

La surabondance de R peut provenir du fait que celui-ci est assez difficile à localiser pour la création de l'ensemble d'apprentissage et que par conséquent il est parfois confondu avec d'autres phonèmes.

Le deuxième résultat présenté concerne aussi le mot stratus, mais en effectuant des itérations tant qu'aucune des probabilités conditionnelles des formes n'a dépassé le seuil de 0,7 sinon le phonème reconnu est celui

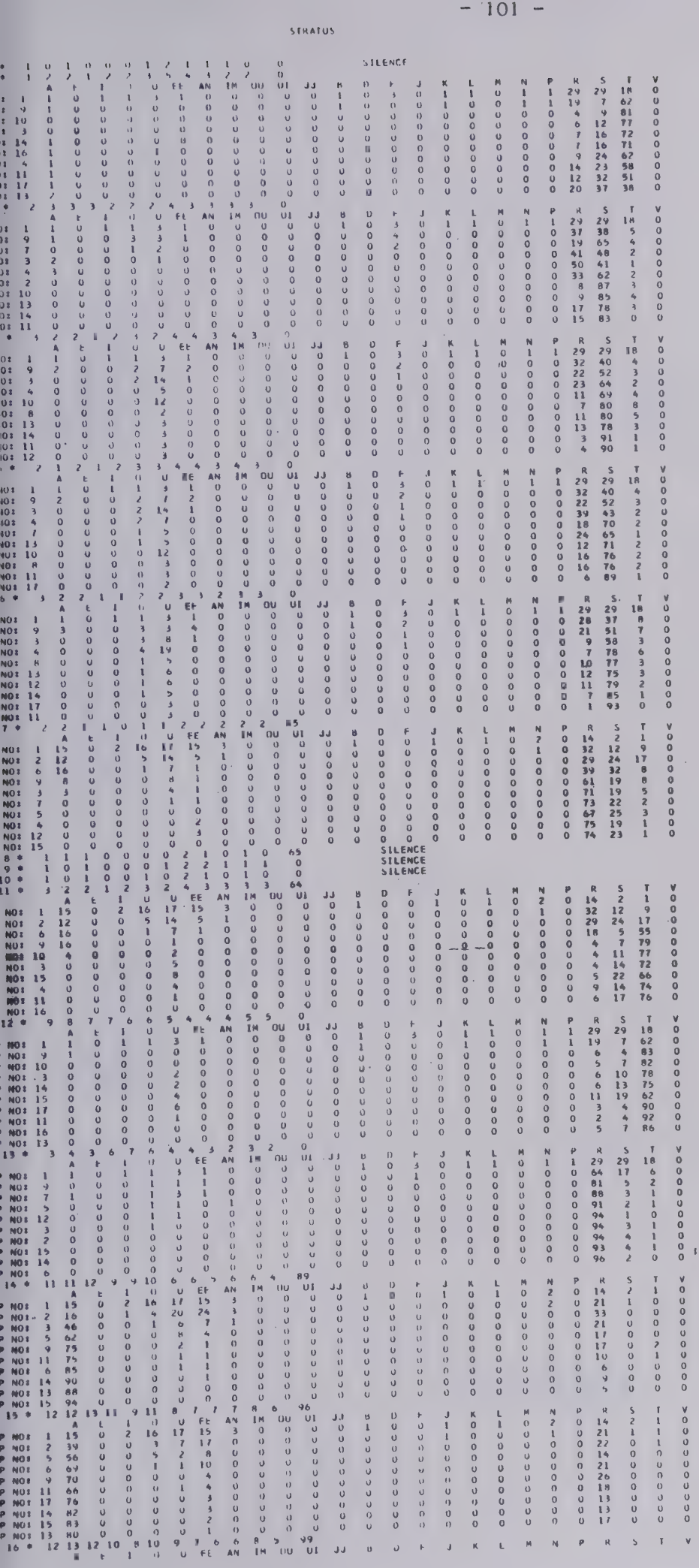


Figure 16 :

Reconnaissance de phonèmes.
Résultats de l'analyse séquentielle lors de la prononciation du mot STRATUS.

STRATUS

1	*	1	0	1	0	0	1	1	2	1	1	1	0	0	
2	*	1	2	2	1	2	2	3	5	4	3	2	2	0	PHONEME
3	*	2	3	3	3	3	2	2	4	3	3	3	3	0	PHONEME
4	*	3	2	2	1	2	3	2	4	4	3	4	3	0	PHONEME
5	*	2	1	2	1	2	3	3	4	4	3	4	3	0	PHONEME
6	*	3	2	2	1	1	2	2	3	3	2	3	3	0	PHONEME
7	*	2	2	1	1	0	1	1	2	2	2	2	2	45	PHONEME
8	*	1	1	1	0	0	0	0	2	1	0	1	0	65	
9	*	1	0	1	0	0	0	1	2	2	1	1	1	0	
10	*	1	0	1	0	0	1	0	2	1	0	1	0	0	
11	*	3	2	2	1	2	3	2	4	3	3	3	3	64	
12	*	9	6	7	7	6	6	5	4	4	4	5	5	0	PHONEME
13	*	3	4	3	6	7	6	4	4	3	2	3	2	0	PHONEME
14	*	11	11	12	9	9	10	6	6	5	6	6	4	89	PHONEME
15	*	12	12	13	11	9	11	8	7	7	7	8	6	96	PHONEME
16	*	12	13	12	10	8	10	9	7	6	6	8	5	99	PHONEME
17	*	10	10	7	6	5	6	6	4	2	2	4	2	0	PHONEME
18	*	5	7	6	4	3	4	2	3	2	2	2	1	0	PHONEME
19	*	5	6	8	3	3	3	2	2	2	2	2	1	103	PHONEME
20	*	5	6	7	3	2	3	2	2	2	1	2	1	50	PHONEME
21	*	3	3	5	2	2	3	2	4	2	1	3	3	0	PHONEME
22	*	7	5	3	4	3	4	4	5	4	5	6	6	0	
23	*	10	7	3	4	4	4	4	8	2	6	7	7	84	PHONEME
24	*	12	9	0	3	3	4	5	9	9	8	5	6	82	
25	*	12	8	4	3	3	3	5	9	3	7	6	6	80	PHONEME
26	*	11	9	5	3	2	3	5	8	4	6	7	7	79	PHONEME
27	*	5	6	6	5	4	3	4	7	5	6	7	8	64	
28	*	7	6	5	4	4	4	4	6	5	6	7	8	0	PHONEME
29	*	7	6	5	4	4	4	4	6	6	7	7	8	0	PHONEME
30	*	8	6	5	3	4	4	4	5	5	6	7	8	0	PHONEME
31	*	6	6	5	4	4	4	4	5	5	5	5	8	0	PHONEME
32	*	5	4	3	4	4	4	4	5	4	5	6	7	0	PHONEME
33	*	4	4	4	3	4	3	3	4	3	4	6	7	0	PHONEME
34	*	4	3	4	3	3	3	4	4	3	4	5	7	0	PHONEME
35	*	5	5	4	4	4	3	3	4	4	4	5	7	0	
36	*	6	4	3	3	2	2	4	4	3	3	4	6	0	PHONEME
37	*	4	4	3	2	1	1	3	3	2	3	2	4	0	PHONEME
38	*	2	1	1	1	0	1	2	2	2	2	1	1	0	PHONEME
39	*	2	1	1	0	0	1	1	2	2	1	1	1	0	PHONEME
40	*	1	0	1	0	0	1	1	2	2	0	1	0	0	

Figure 17-a

```
SILENCE
 RECONNU :   T     1- 9-10-
 RECONNU :   S     1- 9- 7- 3- 4- 2-10-
 RECONNU :   S     1- 9- 3- 4-10- 8-
 RECONNU :   S     1- 9- 3- 4- 7-13-10- 8-11-17-
 RECONNU :   S     1- 9- 3- 4- 8-13-12-14-17-
 RECONNU :   S     1- 2- 6- 9- 3- 7- 5- 4-12-15-16-10-11-
SILENCE
SILENCE
SILENCE

 RECONNU :   T     1- 9-10-
 RECONNU :   R     1- 9- 7-
 RECONNU :   A     1- 2- 3- 5- 9-11- 6-
 RECONNU :   A     1- 2- 5- 6- 9-11-17-14-
 RECONNU :   A     1- 2- 3- 6- 8-15- 5-17-11-
 RECONNU :   R     1- 9- 7-
 RECONNU :   R     1- 9- 7- 2- 5-
 RECONNU :   R     1- 2- 5- 6- 3- 7-
 RECONNU :   R     1- 2- 6- 4- 3- 5-17- 7-
 RECONNU :   R     1- 9- 3- 4- 5- 7- 2- 6-12-15-

 RECONNU :   R     1- 2- 5- 6-15- 8-16- 4- 3-10-11- 9- 7-12-13-14-17-

 RECONNU :   U     1- 2- 3- 6-12-15-
 RECONNU :   U     1- 2- 3- 6-12-14- 7-16-

 RECONNU :   S     1- 9- 3- 4- 2-
 RECONNU :   S     1- 9- 3- 4-13-10-11- 6-17-14- 7- 2-16-
 RECONNU :   S     1- 9- 3- 4-13-10-11- 8-17- 6-
 RECONNU :   S     1- 9- 3- 7- 4-10- 6-13-11-
 RECONNU :   S     1- 9- 3- 4- 8-13- 5- 7-
 RECONNU :   S     1- 9- 3- 6- 4- 7- 2-11-
 RECONNU :   S     1- 9- 3- 4- 7-10- 6-11-13-17-

 RECONNU :   S     1- 9- 7- 3-13- 4-12-11-10-17-14- 6-15- 8-16-
 RECONNU :   R     1- 9- 7- 4- 2-11- 6-10- 3- 5-12-15-
 RECONNU :   S     1- 9- 7- 2- 4-11- 6-10-13- 3-12-17-14-
 RECONNU :   S     1- 9- 3- 4- 2-
SILENCE
```

Figure 17-b

dont la probabilité a dépassé le seuil.

On voit sur cette planche qu'à l'endroit du 2ème T de STRATUS, un R a été reconnu après que les 17 opérateurs aient été utilisés ; on a pris alors R correspondant à la plus forte probabilité.

IV.5. CAUSES D'ERREUR DE LA METHODE

Sur les résultats concernant le mot stratus, que l'on vient de présenter, on trouve des erreurs dans la reconnaissance des phonèmes (sur le T et les R). Toutefois, si nous avions un programme de reconnaissance de mots du type du paragraphe IV.2. dans le contexte des mots du français, le mot stratus aurait été reconnu (redondance au niveau du mot).

Si on veut pourtant encore plus de précision sur la reconnaissance des mots, il serait avantageux d'avoir un algorithme plus fiable de reconnaissance de phonèmes.

Les causes d'erreur de la méthode séquentielle sont par ordre d'importance :

1°) Les opérateurs considérés ne sont pas indépendants. En effet, au lieu d'utiliser la formule de Bayes de manière rigoureuse, elle est utilisée de manière séquentielle (formule 22 : approximation n° 2). On suppose aussi (approximation n° 1) que

$$p[a_j,(k') \mid \omega(i), a_j(k)] = p[a_j,(k') \mid \omega(i)]$$

ce qui est vrai si les opérateurs sont indépendants. Voir IV.1.

2°) L'ensemble des opérateurs heuristiques trouvé peut être insuffisamment informant, en particulier ici le vocoder lui-même.

3°) Le nombre d'échantillons d'apprentissage est trop faible

Les fréquences ne représentent pas les probabilités. C'est le problème de l'ergodicité. de l'ensemble des données et de la représentativité de cet ensemble par l'ensemble d'apprentissage.

Nous avons suivi les hypothèses de travail de NILSSON [53] et nous avons mesuré, lorsque c'était possible, les performances de l'algorithme obtenu par son information utile ou son taux d'erreur sur des données de test différentes des données d'apprentissage.

Dans le problème précédemment étudié de la reconnaissance des phonèmes, les données d'apprentissage avaient été choisies aléatoirement à partir de l'ensemble des données dont nous disposions. Les données de test étaient les données qui restaient.

Laveen KANAL [35] étudie de manière plus précise ces méthodes de choix d'ensembles d'apprentissage et de test. La méthode que nous avons utilisée est parmi les plus simples.

CHAPITRE V

METHODE DE GENERATION AUTOMATIQUE D'OPERATEURS PAR GENERATION ALEATOIRE ET SELECTION PROGRESSIVE

V.1. GENERATION AUTOMATIQUE D'OPERATEURS DE PRETRAITEMENT

Nous avons décrit en III.8.1. une expérience de calcul de l'information utile apportée par un capteur optique en une de la reconnaissance des formes élémentaires {blanc, noir, bord}. Les opérateurs Δ_j considérés étaient au nombre 49, entourant le point du maillage où devait se porter la décision :

$$\{\Delta_j \mid 1 \leqslant j \leqslant 49\}$$

Le meilleur opérateur était l'opérateur central avec une information utile de 65 (à comparer avec l'information utile idéale de 143). Un traitement était donc nécessaire : il fallait créer un opérateur travaillant sur les résultats D_j des Δ_j, le plus proche de l'idéal possible.

La méthode de création d'opérateur employée est un procédé itératif faisant évoluer une liste $\mathcal{L}$ d'opérateurs O. Cette liste est représentée en ordinateur par la liste des symboles des opérateurs et est ordonnée suivant les valeur des $I[A ; \Omega]$ décroissantes. La liste de départ $\mathcal{L}_0$ est faite des 49 opérateurs Δ_j.

1°) Deux opérateurs sont pris au hasard dans la liste, leurs résultats additionnés. Le nouvel opérateur ainsi créé est inséré dans $\mathcal{L}$ à la place correspondant à son information. Le dernier de la liste est

supprimé. Après un nombre d'itérations correspondant à 2 minutes de calcul pour une Univac 1 108, l'opérateur O_1 en tête de liste est décrit par la figure 18.

	2					
		1			1	
					2	
	1			1		
			2	1		

Figure 18

La table précédente donne les coefficients α_j tels que

$$O_1 : X \to A_1 = \sum_j \alpha_j D_j$$

Son information : $I[A_1 ; \Omega] = 103$.

2°) Les résultats des 11 premiers opérateurs de $\mathcal{L}_0$, sont additionnés deux à deux. D'où 55 nouveaux opérateurs. La liste de départ est maintenant de 104 éléments (55 + 49). En appliquant le même procédé qu'au 1°), après 120 itérations l'opérateur O_2 est obtenu en tête de liste :

		1		1	2	
		2			2	
		2	1	1		

Figure 19

La table précédente donne les α_j tels que

$$O_2 : X \to A_2 = \sum_j \alpha_j D_j$$

Son information : $I[A_2 ; \Omega] = 107$.

3°) Nous avons utilisé la multiplication au lieu de l'addition avec le même procédé que le 2°). On obtient :

$$O_3 : X \to A_3 = \prod_j D_j^{\alpha_j}$$

Les α_j sont donnés par le tableau suivant :

		1		1	1	
		2	3		1	
		1	1			
			2	1		

Figure 20

Son information : $I[A_3 ; \Omega] = 75$.

Ses performances sont donc à peine meilleures que celles de la cellule centrale seulement O_{25} : $I[A_{25} ; \Omega] = 65$.

4°) Nous avons ensuite essayé la même méthode mais en associant les opérateurs soit avec le signe plus, soit avec le signe moins (choix aléatoire).

Il est apparu que le signe moins n'apporte aucune amélioration : il n'apparaissait dans le classement final que dans des opérateurs situés après le 17ème meilleur dans la liste.

5°) En utilisant la même liste initiale qu'au 2°) , on associe les opérateurs, non plus aléatoirement mais tels que la somme :

$$I[A_{j_1} ; \Omega] + I[A_{j_2} ; \Omega]$$

est maximum, O_{j_1} et O_{j_2} n'ayant pas été sélectionnés auparavant.

Le nouvel opérateur fait l'addition des deux attributs A_{j_1} et A_{j_2}. La table suivante montre les coefficients α_j de l'opérateur O_5 obtenu en tête de liste après 120 itérations.

					1	
			1	1	1	
		2	3	2	1	
	1	1	2			

Figure 21

L'information $I[A_5 ; \Omega]$ vaut 112.

Cet opérateur O_5 est semblable au filtre digital qu'un expérimentateur aurait proposé après avoir examiné le problème proposé, cf. CHECROUN [11].

On a ensuite testé cet opérateur. La table suivante montre les fréquences d'occurence des évènements conjoints :

$$[\omega(i), a_5(k)]$$

sur l'ensemble d'apprentissage :

Ω \ a_5	0	1	3	7	10	12	13	18	21	23	28	29	32	33	35	36
bord		8	6	10	8	2	17	13	13	13		4	2	3		1
noir						2				1	10	9	12	10	25	41
blanc	69	15	8		2	6										
a_c							•							N		

L'examen de cette table conduit à choisir des seuils de décision entre 3 et 7 et entre 23 et 28. Un nouvel opérateur O_c d'attributs $\{a_c\}$ peut être mis à la suite de O_5. Les symboles représentatifs des a_c sont { , . , N } et sont mis en correspondance avec les notions {blanc ; bord ; noir}.

L'opérateur $O_c \circ O_5$ est une approximation de O_{id}. Le taux d'erreur par rapport à O_{id} est près de 11 %. Cet opérateur a été testé sur un ensemble de test différent de l'ensemble d'apprentissage. Les résultats sont donnés figure 22. La figure 22-a montre les données de départ (sorties du digitaliseur), la figure 22-b montre les résultats de $O_c \circ O_5$ suivant toutes les positions possibles sur la rétine. Les résultats sont satisfaisants.

Remarquons (figure 10) que les données d'apprentissage qui ont permis la génération de cet opérateur étaient entachées d'erreur : deux carrés ont été désignés par le "professeur" comme étant d'abord du blanc et ensuite du noir. L'apprentissage a conduit cependant à un bon opérateur car la probabilité de cette erreur était trop faible pour qu'elle ait une influence notable sur l'opérateur résultant.

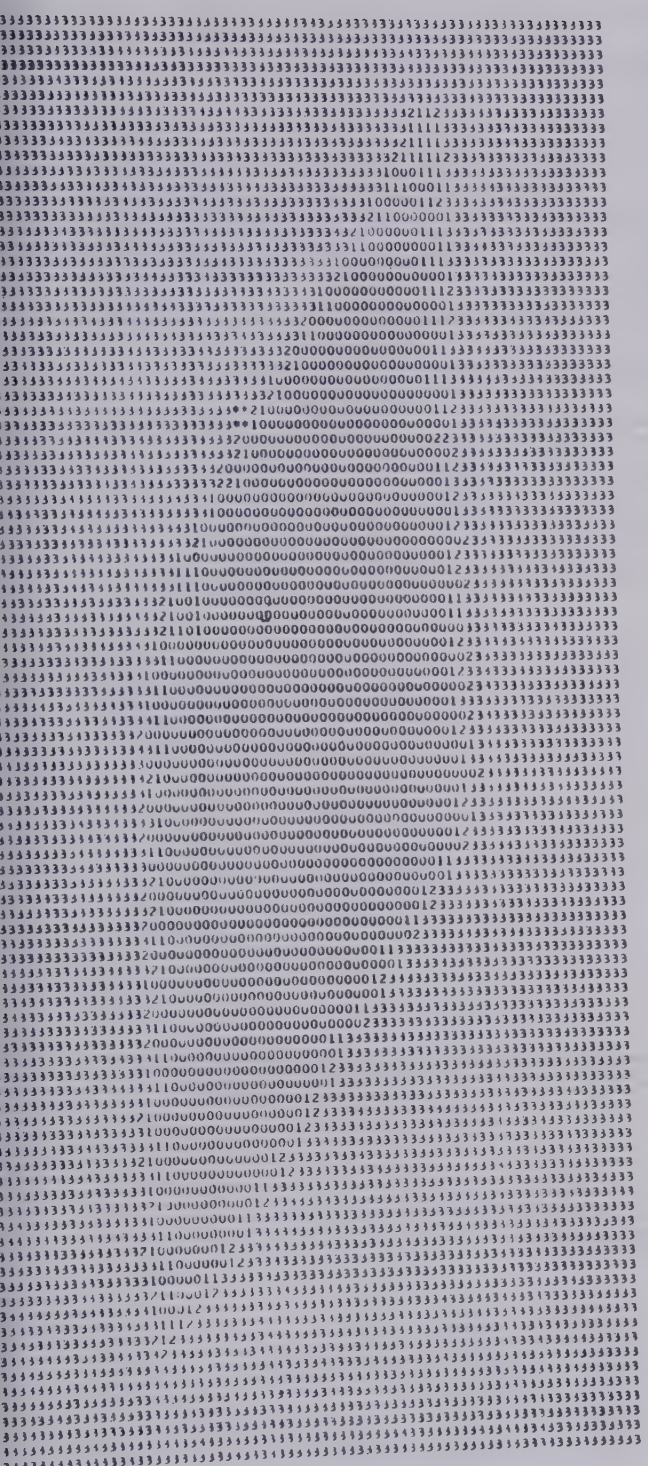


Figure 22-a

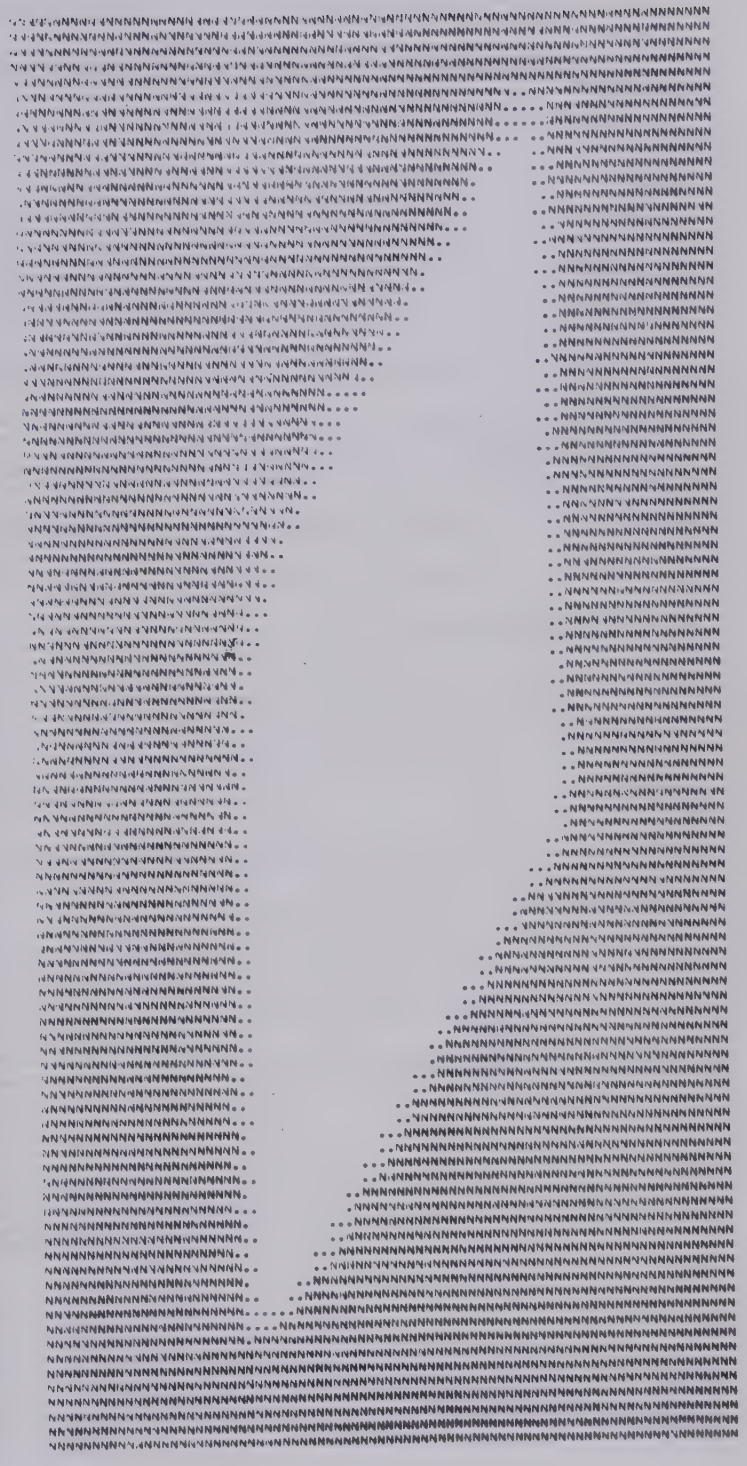


Figure 22-b

V.2. GENERATION AUTOMATIQUE D'OPERATEURS TRAVAILLANT SUR DES RESULTATS D'OPERATEURS DE COMMANDE (OPERATEUR DISTANCE EUCLIDIENNE)

Nous avons voulu utiliser le même programme dans un autre ensemble de données, dans le but d'engendrer un opérateur qui ne soit plus linéaire. Ces expériences ont été décrites dans [63].

L'ensemble des données utilisées est deux ensembles de matrices. Chacune représente une constellation de deux points dont les positions sont symbolisées par deux valeurs 1 dans la matrice dont les autres éléments valent 0. Voir II.2.1. où on considère le même ensemble de données.

Les matrices sont de dimension 9 x 9. Chaque ensemble de matrices en contient 200.

Dans le premier ensemble les deux points sont apparemment "éloignés" l'un de l'autre. Dans le deuxième les points sont apparemment "près" l'un de l'autre.

Dans chaque ensemble, la position du segment joignant les deux points varie en rotation et translation, les notions Ω = {près, éloignés} étant seules conservées dans chaque ensemble.

Conformément au paragraphe II.2.1., nous avons supposé que des opérateurs de commande ont extrait de chaque matrice de données, les coordonnées

X et Y du premier point
U et V du deuxième point

Nous avons fait avec le même programme les mêmes expériences que celles décrites au paragraphe précédent. Nous avons considéré les valeurs X, Y, U et V comme les réponses des opérateurs élémentaires

$$\{O_j \mid j \in 1, 2, 3, 4\}$$

et les formes à reconnaître : Ω = {près, éloigné}.

Une autre méthode de génération donne de bons résultats :

A chaque itération, nous prenons les cinq meilleurs opérateurs de la la liste $\mathcal{L}_n$, nous créons alors les opérateurs correspondant à l'association deux à deux de leurs attributs $(A_j, A_{j'})$ avec tous les signes dont nous disposons (+, - et x). D'où à chaque itération $3 \times 15 = 45$ nouveaux opérateurs (dans l'association $A_j, A_{j'}$, j peut être égal à j'). En ajoutant à ces 45 opérateurs, les 5 meilleurs opérateurs de départ de la liste $\mathcal{L}_n$, on obtient les 50 opérateurs de la liste $\mathcal{L}_{n+1}$. La figure 23 montre les trois premières itérations du procédé : les listes $\mathcal{L}_n$ ont été triées et leurs éléments rangés par ordre d'information utile décroissante.

L'opérateur à atteindre doit avoir une information de 120 (en unités arbitraires) correspondant à la valeur de $H[\Omega]$.

Sur la figure 23, les opérateurs sont symbolisés par leur représentation en notation polonaise post fixée.

L'algorithme s'arrête lorsqu' à une itération, il existe en tête de liste un opérateur dont l'information est 120.

A la première itération nous avons en tête :

$$Y - U,\ X - U,\ Y \cdot V,\ X \cdot U,\ V \cdot V,\ \ldots$$

A la deuxième :

$$(Y - V) - (X - U),\ (Y - V) + (X - U),\ (Y - V) \cdot (X - U),$$
$$(Y - V)^2,\ (X - U)^2,\ \ldots$$

A la troisième enfin :

$$(Y - V)^2 + (X - U)^2,\ ((Y - V) + (X - U))^2,\ ((Y - V) - (X - U))^2,\ \ldots$$

```
VV-      INFORMATION=  9   NOMBRE DE REPONSES :   9
XU-      INFORMATION=  8   NOMBRE DE REPONSES :   9
YV*      INFORMATION=  3   NOMBRE DE REPONSES :  32
XU*      INFORMATION=  2   NOMBRE DE REPONSES :  32
VV*      INFORMATION=  1   NOMBRE DE REPONSES :  10
VV+      INFORMATION=  1   NOMBRE DE REPONSES :  10
YV+      INFORMATION=  1   NOMBRE DE REPONSES :  16
YY*      INFORMATION=  1   NOMBRE DE REPONSES :  10
YY+      INFORMATION=  1   NOMBRE DE REPONSES :  10
 V-      INFORMATION=  1   NOMBRE DE REPONSES :  10
 V+      INFORMATION=  1   NOMBRE DE REPONSES :  10
 Y-      INFORMATION=  1   NOMBRE DE REPONSES :  10
 Y+      INFORMATION=  1   NOMBRE DE REPONSES :  10
V        INFORMATION=  1   NOMBRE DE REPONSES :  10
Y        INFORMATION=  1   NOMBRE DE REPONSES :  10
VV-      INFORMATION=  0   NOMBRE DE REPONSES :   1
JV*      INFORMATION=  0   NOMBRE DE REPONSES :  40
UV-      INFORMATION=  0   NOMBRE DE REPONSES :  18
UV+      INFORMATION=  0   NOMBRE DE REPONSES :  18
UU*      INFORMATION=  0   NOMBRE DE REPONSES :  10
JJ-      INFORMATION=  0   NOMBRE DE REPONSES :   1
UU+      INFORMATION=  0   NOMBRE DE REPONSES :  10
YJ*      INFORMATION=  0   NOMBRE DE REPONSES :  40
YU-      INFORMATION=  0   NOMBRE DE REPONSES :  19
YU+      INFORMATION=  0   NOMBRE DE REPONSES :  18
YY-      INFORMATION=  0   NOMBRE DE REPONSES :   1
XV*      INFORMATION=  0   NOMBRE DE REPONSES :  41
XV-      INFORMATION=  0   NOMBRE DE REPONSES :  17
XV+      INFORMATION=  0   NOMBRE DE REPONSES :  18
XU+      INFORMATION=  0   NOMBRE DE REPONSES :  16
XY*      INFORMATION=  0   NOMBRE DE REPONSES :  41
XY-      INFORMATION=  0   NOMBRE DE REPONSES :  18
XY+      INFORMATION=  0   NOMBRE DE REPONSES :  18
XX*      INFORMATION=  0   NOMBRE DE REPONSES :  10
XX-      INFORMATION=  0   NOMBRE DE REPONSES :   1
XX+      INFORMATION=  0   NOMBRE DE REPONSES :  10
 V*      INFORMATION=  0   NOMBRE DE REPONSES :   1
 J*      INFORMATION=  0   NOMBRE DE REPONSES :   1
 U-      INFORMATION=  0   NOMBRE DE REPONSES :  10
 U+      INFORMATION=  0   NOMBRE DE REPONSES :  10
 Y*      INFORMATION=  0   NOMBRE DE REPONSES :   1
 X*      INFORMATION=  0   NOMBRE DE REPONSES :   1
 X-      INFORMATION=  0   NOMBRE DE REPONSES :  10
 X+      INFORMATION=  0   NOMBRE DE REPONSES :  10
  *      INFORMATION=  0   NOMBRE DE REPONSES :   1
  -      INFORMATION=  0   NOMBRE DE REPONSES :   1
  +      INFORMATION=  0   NOMBRE DE REPONSES :   1
U        INFORMATION=  0   NOMBRE DE REPONSES :  10
X        INFORMATION=  0   NOMBRE DE REPONSES :  10
         INFORMATION=  0   NOMBRE DE REPONSES :   1

 * TEMPS ECOULE  0H.  0MN. 16SEC. 704MSEC. **
```

Figure 23-a : Génération de la distance euclidienne (1ère itération)

```
YV-XJ--        INFORMATION=  10   NOMBRE DE REPONSES :   11
YV-XJ-+        INFORMATION=  30   NOMBRE DE REPONSES :   11
YV-XJ-*        INFORMATION=  24   NOMBRE DE REPONSES :    5
YV-YV-*        INFORMATION=  14   NOMBRE DE REPONSES :    5
XJ-XJ-*        INFORMATION=  13   NOMBRE DE REPONSES :    5
YV-YV-+        INFORMATION=   9   NOMBRE DE REPONSES :    9
YV-            INFORMATION=   9   NOMBRE DE REPONSES :    9
XU-XU-+        INFORMATION=   8   NOMBRE DE REPONSES :    9
XJ-            INFORMATION=   8   NOMBRE DE REPONSES :    9
XU-YV*-        INFORMATION=   4   NOMBRE DE REPONSES :   55
YV-XJ*+        INFORMATION=   4   NOMBRE DE REPONSES :   58
YV*YV**        INFORMATION=   3   NOMBRE DE REPONSES :   32
YV*YV*+        INFORMATION=   3   NOMBRE DE REPONSES :   32
XU-YV*+        INFORMATION=   3   NOMBRE DE REPONSES :   57
YV-XJ*-        INFORMATION=   3   NOMBRE DE REPONSES :   59
YV*            INFORMATION=   3   NOMBRE DE REPONSES :   32
XJ*XJ**        INFORMATION=   2   NOMBRE DE REPONSES :   32
XU*XU*+        INFORMATION=   2   NOMBRE DE REPONSES :   32
YV*VV*+        INFORMATION=   2   NOMBRE DE REPONSES :   48
XU-YV**        INFORMATION=   2   NOMBRE DE REPONSES :   64
YV-XJ**        INFORMATION=   2   NOMBRE DE REPONSES :   67
XU*            INFORMATION=   2   NOMBRE DE REPONSES :   32
VV*VV**        INFORMATION=   1   NOMBRE DE REPONSES :   10
VV*VV*+        INFORMATION=   1   NOMBRE DE REPONSES :   10
YV*VV**        INFORMATION=   1   NOMBRE DE REPONSES :   59
YV*VV*-        INFORMATION=   1   NOMBRE DE REPONSES :   37
YV*XJ**        INFORMATION=   1   NOMBRE DE REPONSES :  117
YV*XJ*-        INFORMATION=   1   NOMBRE DE REPONSES :  105
YV*XJ*+        INFORMATION=   1   NOMBRE DE REPONSES :   89
XU-VV**        INFORMATION=   1   NOMBRE DE REPONSES :   74
XU-VV*-        INFORMATION=   1   NOMBRE DE REPONSES :   69
XJ-VV*+        INFORMATION=   1   NOMBRE DE REPONSES :   70
XU-XJ**        INFORMATION=   1   NOMBRE DE REPONSES :   38
XU-XJ*-        INFORMATION=   1   NOMBRE DE REPONSES :   41
XU-XJ*+        INFORMATION=   1   NOMBRE DE REPONSES :   42
YV-VV**        INFORMATION=   1   NOMBRE DE REPONSES :   52
YV-VV*-        INFORMATION=   1   NOMBRE DE REPONSES :   56
YV-VV*+        INFORMATION=   1   NOMBRE DE REPONSES :   53
YV-YV**        INFORMATION=   1   NOMBRE DE REPONSES :   38
YV-YV*-        INFORMATION=   1   NOMBRE DE REPONSES :   41
YV-YV*+        INFORMATION=   1   NOMBRE DE REPONSES :   41
VV*            INFORMATION=   1   NOMBRE DE REPONSES :   10
VV*VV*-        INFORMATION=   0   NOMBRE DE REPONSES :    1
XU*VV**        INFORMATION=   0   NOMBRE DE REPONSES :  149
XU*VV*-        INFORMATION=   0   NOMBRE DE REPONSES :  113
XU*VV*+        INFORMATION=   0   NOMBRE DE REPONSES :  109
XJ*XJ*-        INFORMATION=   0   NOMBRE DE REPONSES :    1
YV*YV*-        INFORMATION=   0   NOMBRE DE REPONSES :    1
XJ-XJ--        INFORMATION=   0   NOMBRE DE REPONSES :    1
YV-YV--        INFORMATION=   0   NOMBRE DE REPONSES :    1

* TEMPS COULE. 0H. 0MN. 12SEC. 630MSEC. **
```

Figure 23-b : Génération de la distance euclidienne (2ème itération)

```
YV-YV-*XU-XU-*+      INFORMATION=100   NOMBRE DE REPONSES :   3
YV-XU-+YV-XU-+*      INFORMATION= 60   NOMBRE DE REPONSES :   6
YV-XU--YV-XU--*      INFORMATION= 50   NOMBRE DE REPONSES :   6
YV-YV-*XU-XU-**      INFORMATION= 49   NOMBRE DE REPONSES :   3
YV-XU-*YV-XU-**      INFORMATION= 49   NOMBRE DE REPONSES :   3
YV-YV-*XU-XU-*-      INFORMATION= 40   NOMBRE DE REPONSES :   7
YV-XU--YV-XU-+*      INFORMATION= 40   NOMBRE DE REPONSES :   7
YV-XU-+YV-XU-*-      INFORMATION= 30   NOMBRE DE REPONSES :  10
YV-XU-+YV-XU-*+      INFORMATION= 30   NOMBRE DE REPONSES :  10
YV-XU-+YV-XU-++      INFORMATION= 30   NOMBRE DE REPONSES :  11
YV-XU--YV-XU-*-      INFORMATION= 30   NOMBRE DE REPONSES :  10
YV-XU--YV-XU-*+      INFORMATION= 30   NOMBRE DE REPONSES :  10
YV-XU--YV-XU--+      INFORMATION= 30   NOMBRE DE REPONSES :  11
YV-XU-+              INFORMATION= 30   NOMBRE DE REPONSES :  11
YV-XU--              INFORMATION= 30   NOMBRE DE REPONSES :  11
YV-XU-*XU-XU-*-      INFORMATION= 25   NOMBRE DE REPONSES :   8
YV-XU-*XU-XU-*+      INFORMATION= 25   NOMBRE DE REPONSES :   8
YV-XU-*YV-YV-*-      INFORMATION= 25   NOMBRE DE REPONSES :   8
YV-XU-*YV-YV-*+      INFORMATION= 25   NOMBRE DE REPONSES :   8
YV-XU-*YV-XU-*+      INFORMATION= 24   NOMBRE DE REPONSES :   5
YV-XU-*              INFORMATION= 24   NOMBRE DE REPONSES :   5
YV-XU-*XU-XU-**      INFORMATION= 16   NOMBRE DE REPONSES :   7
YV-XU-*YV-YV-**      INFORMATION= 16   NOMBRE DE REPONSES :   7
YV-XU-+XU-XU-*-      INFORMATION= 16   NOMBRE DE REPONSES :  18
YV-XU-+YV-YV-*-      INFORMATION= 16   NOMBRE DE REPONSES :  18
YV-XU-+YV-YV-*+      INFORMATION= 16   NOMBRE DE REPONSES :  18
YV-XU-+YV-XU-**      INFORMATION= 16   NOMBRE DE REPONSES :   7
YV-XU--YV-YV-*-      INFORMATION= 16   NOMBRE DE REPONSES :  18
YV-XU--YV-XU-**      INFORMATION= 16   NOMBRE DE REPONSES :   7
YV-XU-+XU-XU-*+      INFORMATION= 15   NOMBRE DE REPONSES :  18
YV-XU--XU-XU-*-      INFORMATION= 15   NOMBRE DE REPONSES :  18
YV-XU--XU-XU-*+      INFORMATION= 15   NOMBRE DE REPONSES :  18
YV-XU--YV-YV-*+      INFORMATION= 15   NOMBRE DE REPONSES :  18
YV-YV-*YV-YV-**      INFORMATION= 14   NOMBRE DE REPONSES :   5
YV-YV-*YV-YV-*+      INFORMATION= 14   NOMBRE DE REPONSES :   5
YV-YV-*              INFORMATION= 14   NOMBRE DE REPONSES :   5
XU-XU-*XU-XU-**      INFORMATION= 13   NOMBRE DE REPONSES :   5
XU-XU-*XU-XU-*+      INFORMATION= 13   NOMBRE DE REPONSES :   5
XU-XU-*              INFORMATION= 13   NOMBRE DE REPONSES :   5
YV-XU-+XU-XU-**      INFORMATION= 11   NOMBRE DE REPONSES :  13
YV-XU-+YV-YV-**      INFORMATION= 10   NOMBRE DE REPONSES :  13
YV-XU--XU-XU-**      INFORMATION= 10   NOMBRE DE REPONSES :  13
YV-XU--YV-YV-**      INFORMATION= 10   NOMBRE DE REPONSES :  13
YV-XU--YV-XU-++      INFORMATION=  9   NOMBRE DE REPONSES :   9
YV-XU--YV-XU-+-      INFORMATION=  8   NOMBRE DE REPONSES :   9
XU-XU-*XU-XU-*-      INFORMATION=  0   NOMBRE DE REPONSES :   1
YV-YV-*YV-YV-*-      INFORMATION=  0   NOMBRE DE REPONSES :   1
YV-XU-*YV-XU-*-      INFORMATION=  0   NOMBRE DE REPONSES :   1
YV-XU-+YV-XU-+-      INFORMATION=  0   NOMBRE DE REPONSES :   1
YV-XU--YV-XU---      INFORMATION=  0   NOMBRE DE REPONSES :   1
          L INFORMATION MAXIMALE (100) A ETE TROUVEE
          ***********************************************)
        * TEMPS ECOULE  0H.  0MN. 13SEC. 715MSEC. **
```

Figure 23-c : Génération de la distance euclidienne (3ème itération)

Le processus s'arrête, puisque l'opérateur trouvé est totalement informant : la distance euclidienne a ainsi été engendrée.

On remarque que les opérateurs les plus informants de la première itération, sont les termes du développement de $(Y - U)^2 \star (X - U)^2$ ou des termes intervenant directement dans le calcul de ce dernier opérateur.

Ce fait ne peut être expliqué par les théorèmes du chapitre III. En effet, étant données des variables A_1 et A_2, les formules :

$$I[A_1, A_2 ; \Omega] \geqslant I[A_1 ; \Omega] + I[A_2 ; \Omega] - I[A_1 ; A_2]$$
$$\geqslant I[A_1 ; \Omega]$$

montrent qu'on ne peut qu'augmenter l'information utile en associant A_1 et A_2. On verra dans le chapitre suivant qu'il existe un opérateur, qui, en utilisant A_1 et A_2 d'entrée transmet une information utile supérieure à celle de ces variables .

On a donc une probabilité non nulle avec un algorithme du type précédent de créer des opérateurs de reconnaissance de plus en plus informants au fur et à mesure des itérations.

On ne peut par contre rien en déduire sur la convergence de l'algorithme vers l'optimum :

$$I[A ; \Omega] = H[\Omega]$$

Dans de très nombreux cas pratiques l'optimum n'est utile que pour une certaine satisfaction de l'esprit, l'utilisateur demande simplement un bon opérateur dont il connaisse parfaitement les performances. La méthode de génération aléatoire et sélection progressive est donc une bonne heuristique pour l'apprentissage d'un tel opérateur.

Dans les cas où on cherche l'optimum, on peut aussi l'utiliser pour engendrer l'opérateur idéal mais sans aucune certitude sur la convergence du procédé.

V.3. CAS PARTICULIER DES OPERATEURS A SEUIL

Nous avons fait des expériences de mesure d'information utile en utilisant le même ensemble d'apprentissage que celui qui nous a conduit au filtre linéaire pour extraire les formes {bord, blanc, noir}.

Elles avaient pour but d'étudier le comportement de l'information utile d'opérateurs dépendant de deux paramètres lorsqu'on faisait varier ceux-ci.

Les opérateurs qui ont été étudiés sont :

1°) $$V = ((D_{11} - D_{77})^2 + (D_{17} - D_{71})^2) \times 40 + (D_{11} + D_{77} + D_{71} + D_{17} + D_{33} + D_{34} + D_{43})$$

- $A = 2$ si $V > S_1$
- $A = 1$ si $S_2 < V \leqslant S_1$
- $A = 0$ si $V \leqslant S_2$

2°) $$V = D_{11} + D_{22} + D_{12} + D_{21} + D_{76} + D_{77} + D_{66} + D_{67} + D_{17} + D_{16} + D_{26} + D_{27} + D_{61} + D_{62} + D_{71} + D_{72}$$

- $A = 2$ si $V > S_1$
- $A = 1$ si $S_2 < V \leqslant S_1$
- $A = 0$ si $V \leqslant S_2$

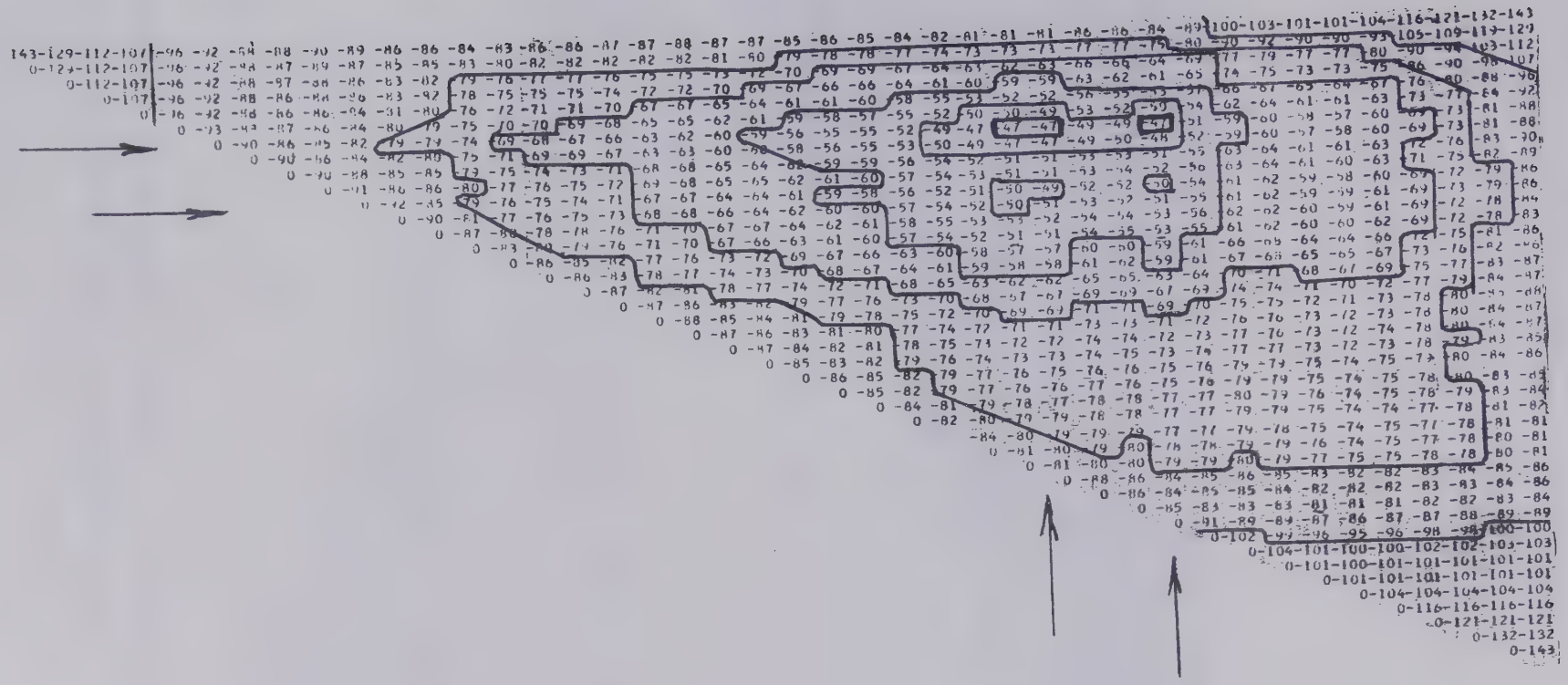


Figure 24 : Opérateurs à 2 seuils - Evolution de l'information utile transmise par le 1er opérateur en fonction des valeurs des 2 seuils.

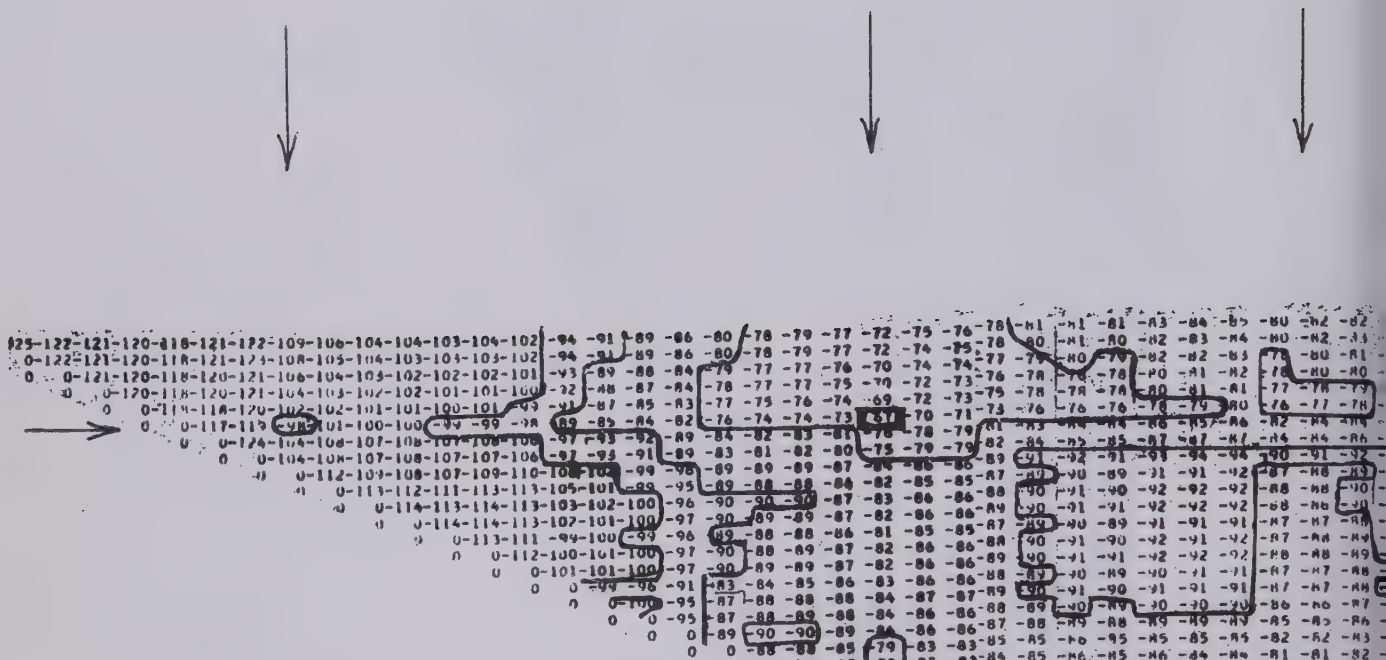
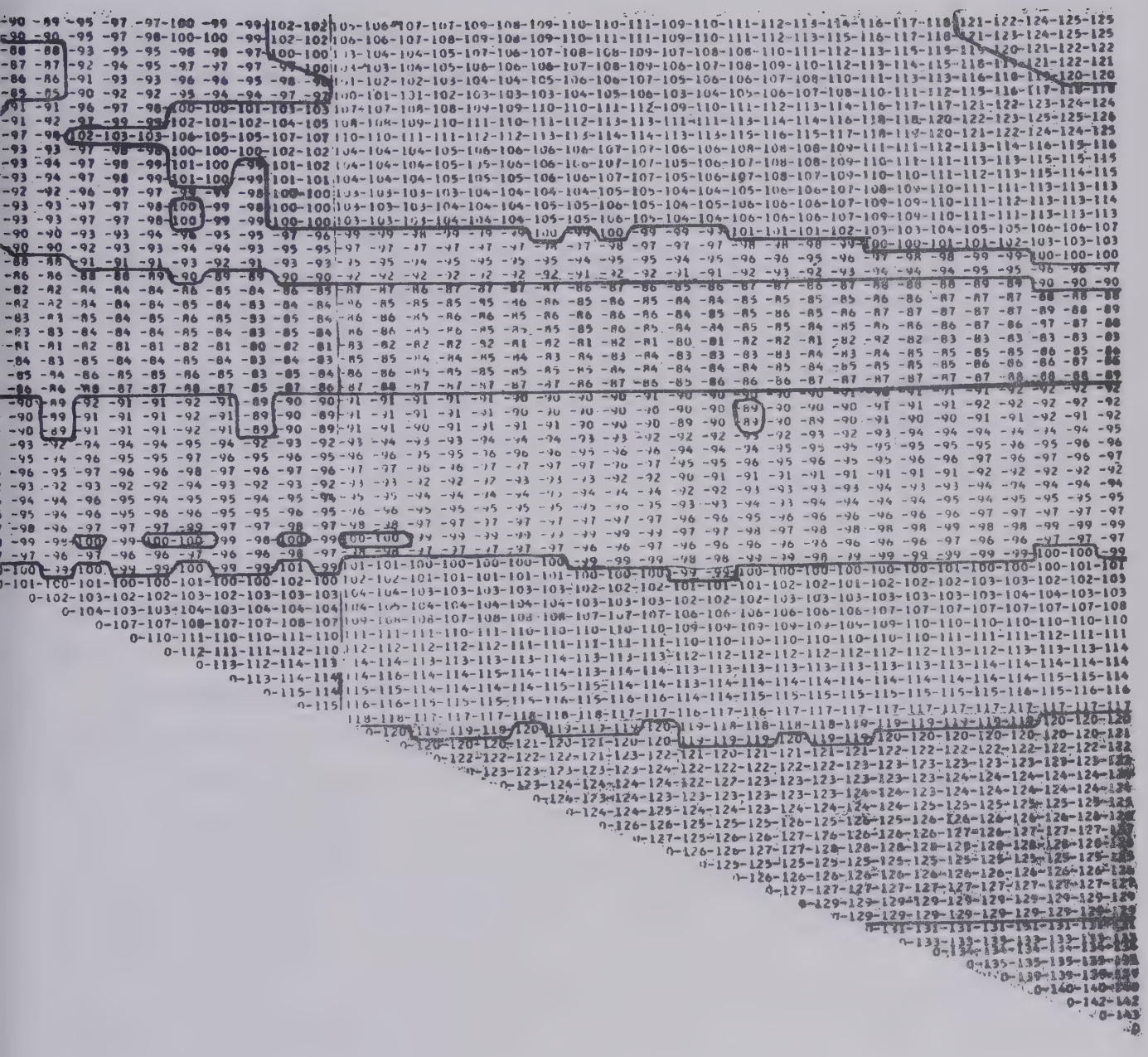


Figure 25 : Opérateurs à 2 seuils - Evolution de l'information utile transmise par le 2ème opérateur en fonction des valeurs des 2 seuils.



Les deux paramètres variables sont dans les deux cas les valeurs S_1 et S_2 des "seuils de décision". Les pages précédentes montrent les valeurs de $-H[\Omega \mid A]$ en fonction des valeurs des deux seuils.

On a donc une manière de déterminer automatiquement les valeurs des paramètres S_1 et S_2. On choisit ces valeurs correspondant au maximum de $I[\Omega ; A]$, ici au maximum de $-H[\Omega \mid A]$ (minimum en valeur absolue).

En généralisant ce procédé, étant donné un opérateur à m paramètres, une recherche utilisant un "Hill-climbing" permet de déterminer automatiquement les valeurs optimales des paramètres.

Dans cet exemple précis on remarque que :

Les valeurs optimales de S_1 et S_2 sont <u>indépendantes</u> l'une de l'autre. Si en effet on fixe l'un des deux paramètres à une valeur quelconque, l'optimum de l'autre est indépendant de cette valeur. Dans ce cas, le hill-climbing converge donc très vite.

C'est ainsi que l'on peut déterminer les deux seuils de décision du paragraphe V.1. Mais dans ce cas nous les avions obtenus par simple examen du tableau des probabilités

$$p[\omega(i), a_5(k)]$$

On voit d'ailleurs sur ce dernier cas la raison pour laquelle les valeurs de S_1 et S_2 sont indépendantes l'une de l'autre : chacun des seuils sépare deux des trois formes à reconnaître.

V.4. CONCLUSION - APPLICATIONS PRATIQUES

Dans chacun des cas présentés le processus de génération a convergé vers un opérateur intéressant. La convergence existe dans les cas où un bon opérateur de reconnaissance est lui-même construit sur des données informantes. Ces cas sont assez fréquents dans la pratique.

Lorsqu'on n'a aucune idée de l'opérateur intéressant, l'ensemble des opérateurs ne peut que s'améliorer par une méthode inspirée des méthodes précédentes. On peut changer les algorithmes de choix d'opérateur suivant les résultats des différentes convergences.

L'ordinateur est utilisé alors comme une aide à la recherche. Il fonctionne moins intuitivement que le chercheur humain, mais peut examiner dans le même temps beaucoup plus de solutions.

Cette utilisation a été rendue possible grâce à la définition d'une fonction d'évaluation des opérateurs, permettant une sélection automatique.

Cette méthode n'est pas sans rappeler le processus de création évoqué par MONOD dans "Le hasard et la nécessité". Nous ne développons pas plus le possible intérêt philosophique de cette corrélation.

Cette méthode est utilisable dans certains problèmes pratiques où d'autres méthodes sont impossibles. Certains autres plus complexes entraînent une convergence trop lente.

Une idée serait d'engendrer automatiquement les opérateurs de choix. Malheureusement, un trop grand ensemble d'apprentissage serait nécessaire : ce qui est possible pour les êtres vivants qui disposent de plusieurs années d'apprentissage est pour l'instant impossible pour un ordinateur dont l'apprentissage ne peut guère dépasser 24 heures... (Nous ne comptons pas dans ce calcul les données héréditaires implantées chez les êtres vivants par les millions d'années de l'évolution...).

Par contre, l'expérience humaine permet d'accélérer le processus d'apprentissage des ordinateurs, par l'introduction d'heuristiques : celles

que nous avons utilisées par exemple pour déterminer précédemment les choix successifs des opérateurs. La notion même d'"information utile" est une heuristique qui nous a servi dans la sélection des meilleurs opérateurs.

CHAPITRE VI

METHODE DE GENERATION AUTOMATIQUE D'OPERATEURS PAR LEUR TABLE

VI.1. PRINCIPES GENERAUX

Nous avons introduit aux chapitres I et II, la notion essentielle d'opérateurs et de structures d'opérateurs. On a vu que les structures d'opérateurs peuvent être considérées comme des opérateurs. Réciproquement, chaque opérateur peut être représenté soit par sa table (eg : la table de l'addition) et les variables sur lesquelles il travaille, soit par une structure d'opérateurs plus simples (l'additon de a et b peut être considérée comme b fois l'addition de a + 1).

La deuxième notion introduite est celle de la factorisation d'opérateurs. Par exemple, au lieu d'avoir b additions (a + 1), on factorise l'addition a + 1 et on l'utilise b fois : on paramètrise l'opérateur A := A + 1 correspondant par un indice m, et on crée un opérateur de commande sur m qui par incrément et test fait travailler l'opérateur global A := A + 1, b fois.

La notion d'opérateur a été introduite, (chapitres suivants) pour diminuer l'information totale dont on dispose dans les données à traiter et pour ne garder que l'information utile pour les traitements ultérieurs.

Par contre, les notions de factorisation et de paramétrisation ont été introduites dans le but exclusif d'occuper pour un traitement donné, le minimum de place en mémoire de l'ordinateur ou le minimum de cablage possible pour le système de traitement de données.

Nous nous sommes posés (chapitre V) le problème d'engendrer des structures d'opérateurs dans des cas précis, lorsque nous ne disposions que d'opérations élémentaires simples tels que l'addition, la soustraction, la multiplication. Certaines méthodes nous ont conduits à la fois à la conception de filtres linéaires et de l'opérateur distance euclidienne. Mais ces méthodes convergent lentement, et surtout :

Dans les cas pratiques, il n'y a aucune raison en général que des opérateurs constitués d'addition, soustraction et multiplication soient adaptés à la reconnaissance des formes.

Remarque - La présence de ces opérations cablées en ordinateur est uniquement due au fait que ces opérations sont intéressantes en comptabilité, en calcul numérique (séries entières), et certains traitements de données numériques, (intérêt du filtre linéaire et calcul des probabilités). Nous les utilisons d'ailleurs nous-mêmes ici puisque la fonction information mutuelle définition de l'information utile, nécessite pour son calcul les opérations addition et multiplication !

Jusqu'à présent ces opérations étaient utiles dans de nombreux cas. Aussi, a-t-on voulu les factoriser avec certaines autres pourqu'elles puissent servir dans de nombreuses applications pratiques : d'où l'ordinateur comme support de ces opérations.

Un ordinateur est en effet un ensemble d'opérations cablées qui peuvent travailler sur un nombre important de variables (mémoire). Les paramètres de ces opérations sont déterminés soit par l'homme (programme) soit par les opérateurs de l'ordinateur lui-même (compilateur, modification d'instruction, programme d'apprentissage d'opérateurs en Reconnaissance des Formes).

L'ordinateur est aussi utile comme outil de conception par ses opérations cablées de gestion de mémoire. En effet, nous pouvons décider de créer des opérations spécialisées autres que l'addition et la multiplication et donc de les créer par leur table de correspondance : les instructions d'indirection de l'ordinateur et les instructions de lecture de tableaux le permettent.

Une fois que de tels opérateurs ont été conçus, on peut alors en faire un cablage spécialisé dans le cas où son utilisation est constante. L'ordinateur peut alors traiter les problèmes généraux, pour lesquels il a été conçu, en particulier les problèmes d'apprentissage et de conception de systèmes.

Nous allons exposer des méthodes de génération d'opérateurs généraux par leur table. Elles nous ont permis de les utiliser dans des structures simples de reconnaissance, utilisant ces opérateurs et, dans certains cas, d'engendrer automatiquement cette structure.

D'habitude, la difficulté principale dans la génération automatique de systèmes de reconnaissance réside dans la génération de cette stucture ainsi qu'il a été fait dans les exemples du chapitre V.

Nous nous sommes permis de contourner cette difficulté en créant des opérateurs élémentaires plus complexes et par là même de simplifier le problème de la génération de la stucture.

La création de ces opérateurs élémentaires spécialisés a été rendue possible

1. par l'introduction d'une fonction d'évaluation de ces opérateurs : l'information utile définie précedemment.

2. par le fait que l'opérateur le plus général peut être défini par ses variables d'entrée et de sortie et par sa table de correspondance

et surtout

3. que la simulation d'un tel opérateur et le calcul de sa fonction d'évaluation sont à la fois possibles dans un ordinateur...

SAMUEL dans son programme de jeu de dames (checkers) a de même utilisé des opérateurs définis par leur table . Il les a créé sans notion d'information utile, en se servant d'heuristiques travaillant sur des données d'apprentissage [69].

Remarque :

La définition en ordinateur d'un opérateur par sa table, n'est pas due à une simple nécessité provenant d'une impossibilité de créer la structure correspondante avec des opérateurs simples. En effet, dans de nombreux cas la méthode du calcul par tableaux est beaucoup plus rapide.

En effet, prenons l'exemple du calcul de l'information mutuelle. Dans chacune des réalisations pratiques présentées, il a fallu calculer celle-ci, donc utiliser l'opérateur :

$$Y \leftarrow \text{Log } (X)$$

Les ordinateurs dont nous disposions calculent cet opérateur par la somme d'une série (structure sur les opérations élémentaires addition et multiplication). Cet opérateur est par ailleurs utilisé un nombre important de fois puisqu'il intervient dans une somme pondérée à plusieurs indices (structure sur les opérations élémentaires addition et multiplication), d'où un temps de calcul prohibitif. Aussi avons nous établi le tableau de la correspondance

$$Y \leftarrow \text{Log } (X)$$

le temps de calcul de cet opérateur est alors le temps d'une simple indirection en ordinateur. Par contre l'occupation mémoire de la table est importante (1 000 valeurs dans nos problèmes).

VI.2. GENERATION DE LA TABLE DE L'OPERATEUR

Nous avons montré au paragraphe I.2.3. comment un opérateur peut-être défini par sa table, que l'opérateur soit arithmétique ou syntaxique. Nous avons montré au paragraphe III.5.3. en prenant comme exemple l'addition que l'information utile apportée par l'opérateur est indépendante des noms a(k) des réponses de l'opérateur mais dépend exclusivement des couples de réponses regroupées par l'action de l'opérateur.

Si les données sur lesquelles travaille l'opérateur considéré sont

$$A_1 = \{a_1(k_1)\} \text{ et } A_2 = \{a_2(k_2)\},$$

l'information utile apportée par l'opérateur dépend exclusivement de la partition qu'il implique sur l'ensemble $(A_1 \otimes A_2)$ produit cartésien des ensembles A_1 et A_2 : elle ne dépend que des valeurs $(a_1(k_1), a_2(k_2))$ qui ont été regroupées par l'action de l'opérateur, sous le même nom, indépendamment de ce nom.

On peut ramener alors le problème de la génération de la table, à la génération de la partition. Nous pouvons par exemple créer la partition $p(A_1 \otimes A_2)$ en n classes qui donne la valeur maximum de

$$I[p(A_1 \otimes A_2) ; \Omega]$$

si notre but est d'avoir un opérateur à n réponses travaillant sur A_1 et A_2 et qui donne l'information utile maximale.

Les méthodes de regroupement (en anglais : "clustering") en vue d'optimiser une certaine fonction, ont été étudiées par de nombreux auteurs dans le cas général. Le problème est ici particularisé. Nous nous sommes en effet imposés des opérateurs :

- travaillant sur deux variables : $z = f(x, y)$
- dont les variables d'entrée peuvent prendre des valeurs discrètes en nombre limité à 16. Nous les avons arbitrairement numérotés de 1 à 16 (leur nom n'a aucune importance).

La limite à 16 du nombre des valeurs différentes de chaque donnée, provient de ce que des données optiques ou acoustiques obtenues avec cette précision, offrent expérimentalement une information suffisante.
Avec 8 niveaux (et même 4) on peut en effet reconstituer convenablement une image photographique.

VI.2.1. Méthode de regroupement employée

Les différentes méthodes étudiées par les auteurs peuvent être employées en tenant compte du compromis entre la rapidité de la convergence et la précision du maximum de la fonction à optimiser [90].

DIDAY propose da méthode des nuées dynamiques : à chaque itération on engendre une partition en prenant les points les plus près des "noyaux" des parties de la partition de l'itération précédente [15],[16]. Une variante simplifiée de cette méthode est la méthode de Mc QUEEN [45]. Celle que nous avons utilisée est très voisine de cette dernière :

A chaque itération la partition obtenue a un élément de moins qu'à l'itération précédente : on regroupe chaque fois deux parties de la partition en une seule. Pour décider des deux parties à regrouper on examine m parties de la partition, on calcule pour les C_m^2 regroupements possibles de deux de ces parties la valeur de la fonction à optimiser et on choisit à chaque pas le meilleur regroupement. C'est-à-dire ici le couple de réponses qui correspond à la baisse d'information la plus faible.

Nous avons essayé cette méthode en utilisant nos données vocales (III.8.2.) en prenant les données d'apprentissage correspondant aux deux canaux du vocoder les plus bas en fréquence. Chacun des canaux O_1 et O_2 donne un résultat entier compris entre 0 et 15. On a donc au départ la partition suivante :

	0			15
0	1	2		16
	17	18		32
	33			
15				256

qui correspond à un opérateur à 256 réponses donnant l'information utile maximum : $I[A ; \Omega] = I[A_1, A_2 ; \Omega]$.

Puis à chaque itération, le nombre de parties diminue d'une unité à partir de 256, jusqu'à ce qu'il n'y en ait plus qu'une. Dans ce dernier cas, l'information utile est nulle, ainsi que l'information totale transmise : quelles que soient les données d'entrée, l'opérateur donne alors toujours la même réponse.

La page suivante montre en fonction du numéro de l'itération l'évolution de l'information utile transmise. Sur le même graphique on montre l'évolution de l'information utile transmise en prenant le plus mauvais regroupement à chaque itération : courbe inférieure.

Les pages suivantes montrent les partitions effectivement obtenues ; les tables qui traduisent ces partitions sont les tables d'opération des opérateurs correspondants. On a étudié les partitions créées en fonction de m : d'abord la valeur de m maximum (le sous-ensemble de la partition sur lequel se fait le choix du regroupement est la partition elle-même toute entière), et ensuite la valeur de m petite (15) correspondant à une convergence rapide. Le temps de convergence est en effet proportionnel à C_m^2.

Les figures présentées montrent que les courbes sont voisines surtout en fin de convergence.

Les tableaux diffèrent jusqu'aux itérations donnant 16 à 14 réponses. Au voisinage de 7 réponses, les partitions sont sensiblement les mêmes.

Nous avons étudié l'influence du choix du sous-ensemble des m parties de la partition, dans l'évolution de l'information utile transmise.

L'algorithme de regroupement de Mc QUEEN implique en ordinateur l'existence d'un vecteur ordonné des noms des parties susceptibles d'être regroupées (en l'occurrence leur numéro dans l'ordre croissant). Chaque fois que deux parties sont regroupées, leur regroupement prend le nom de la première des deux parties, et la dernière des parties du vecteur prend le nom de la deuxième de ces deux parties.

Différents choix de ces m parties, ont été étudiés :

- les m premières de la liste ordonnée
- les m dernières
- lorsqu'on en a considéré m à l'itération précédente, les m suivantes dans l'ordre du vecteur ordonné ; en fin de liste, les m premières.

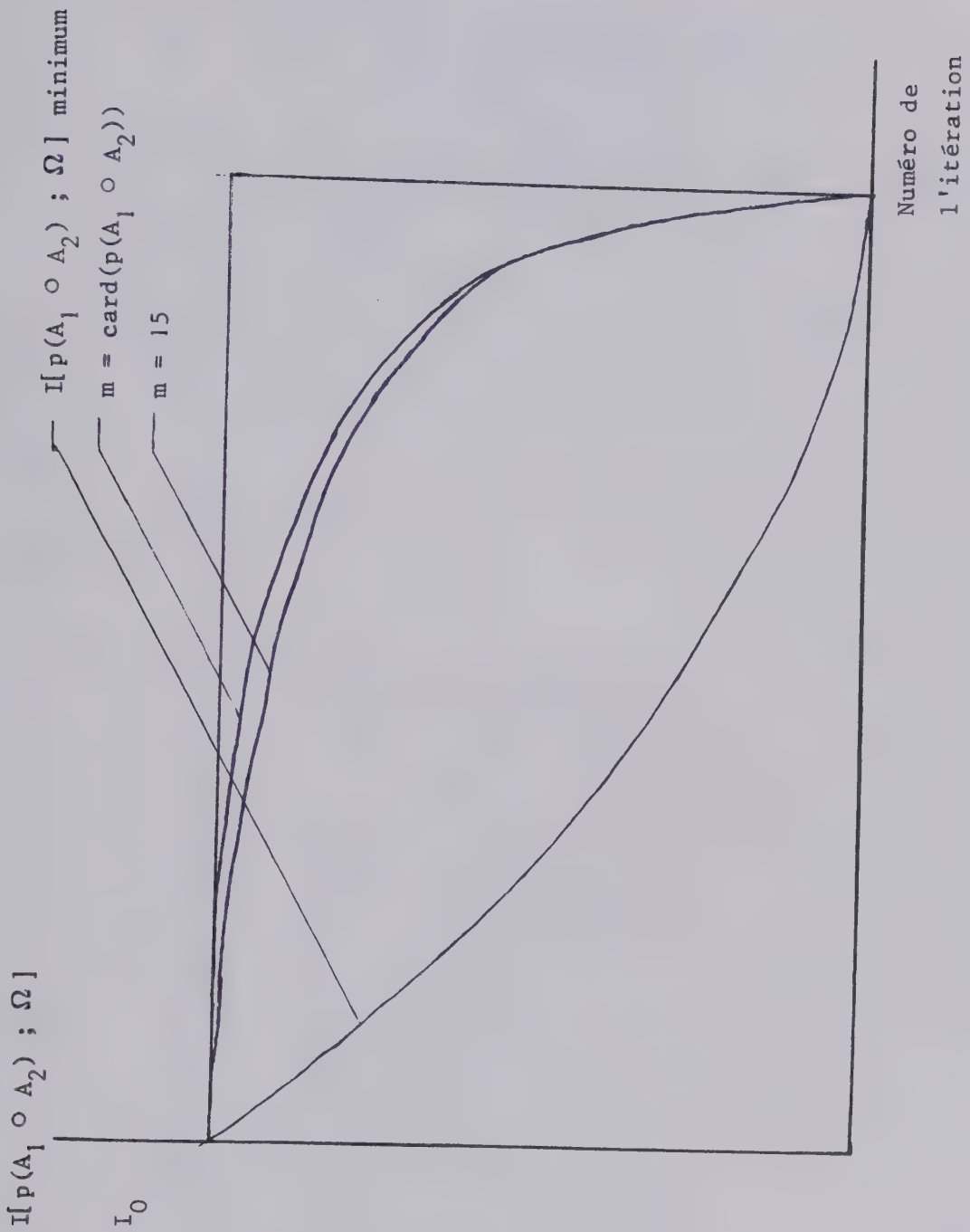


Figure 26 : Variations de l'information mutuelle $I[p(A_1 \circ A_2) ; \Omega]$, en fonction des valeurs de m. Critère : I maximum.

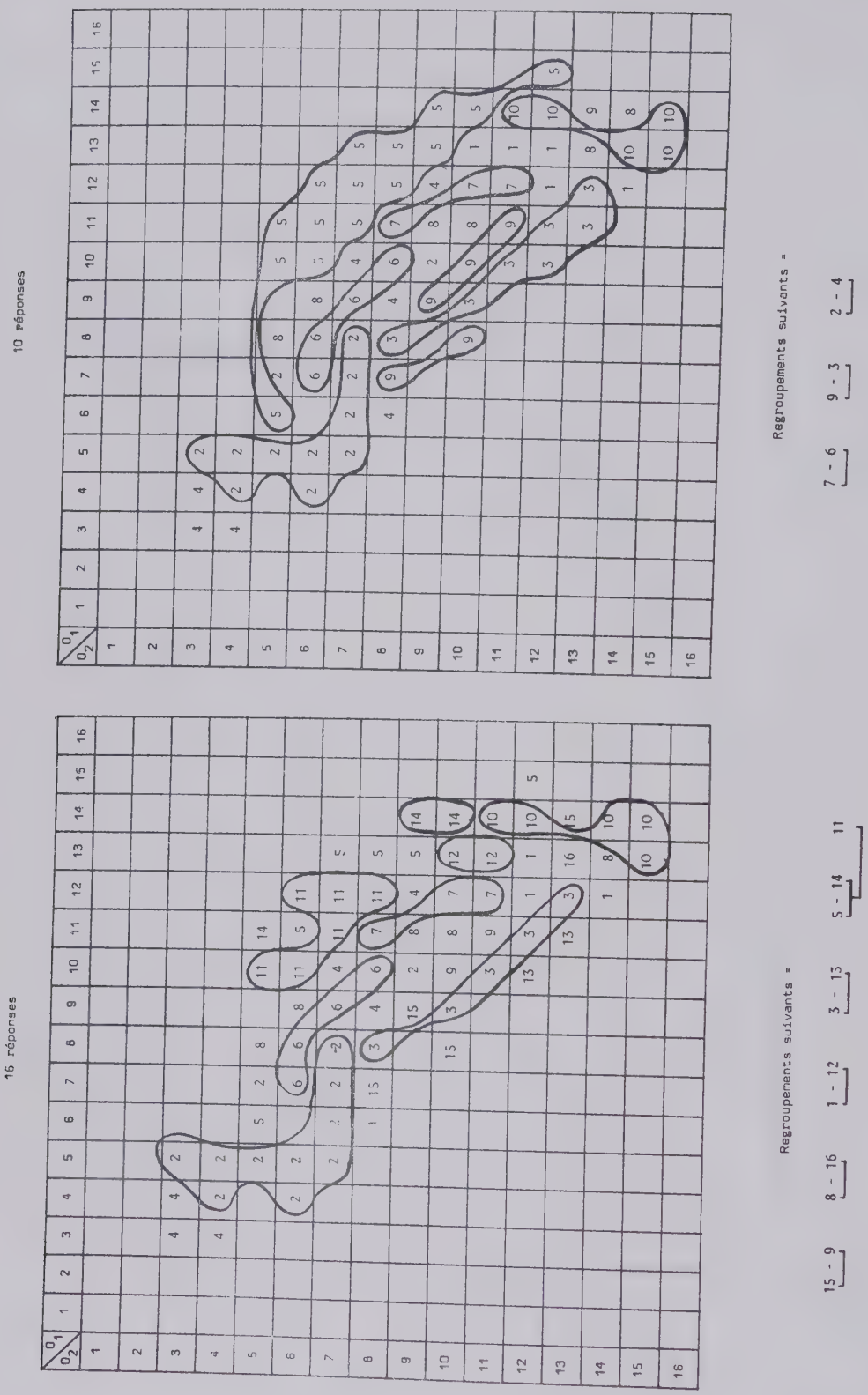


Figure 27-a : Regroupements des réponses de O_1 et O_2
Critère $I[p(A_1 \circ A_2 ; \Omega]$ maximum.
Valeur maximale de m.

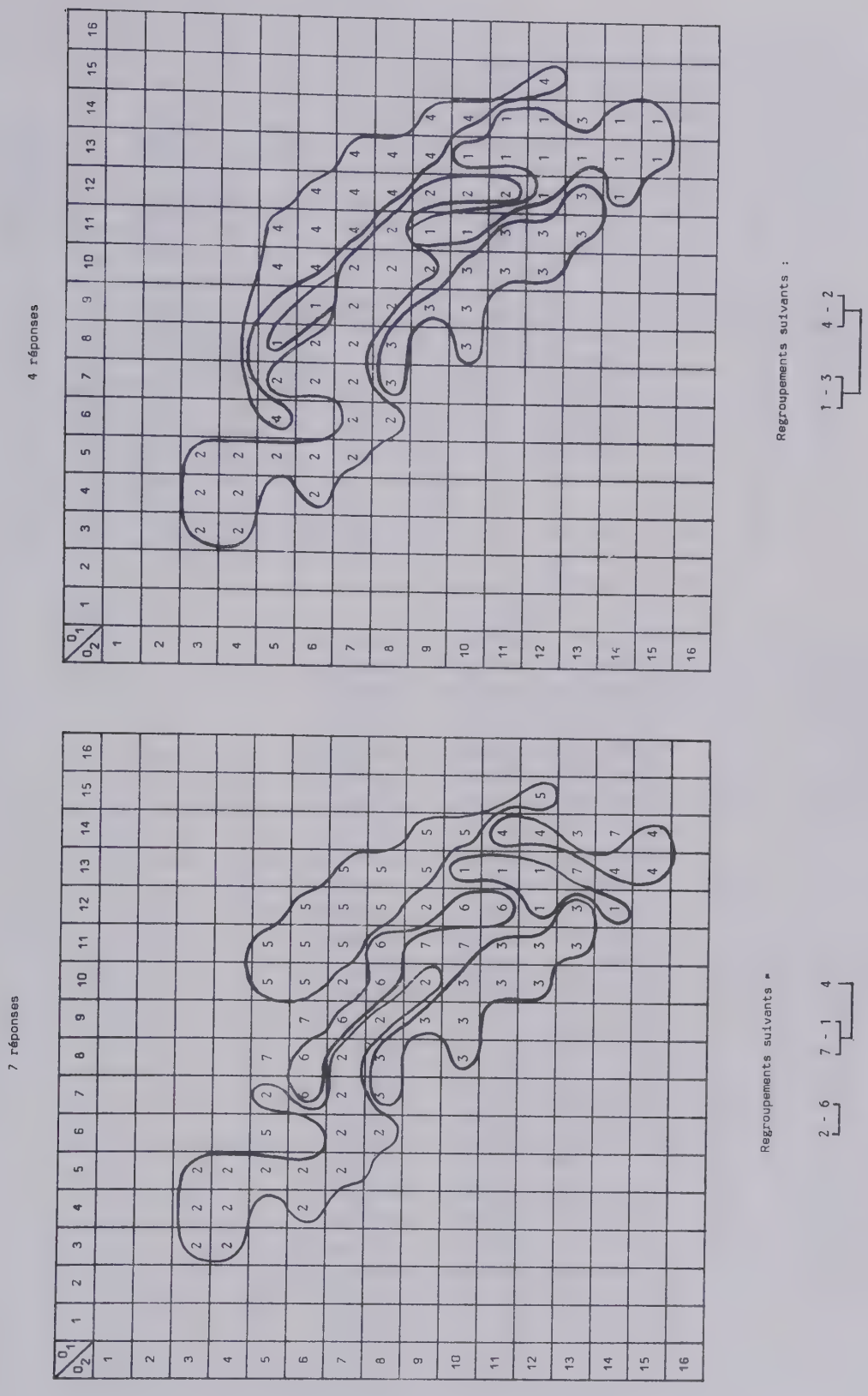


Figure 27-b : Regroupements des réponses de O_1 et O_2
Critère $I[p(A_1 \circ A_2) ; \Omega]$ maximum
Valeur maximale de m.

```
RUN
REGROUPEMENT DES RESULTATS DES OPERATEURS NO  5 ET  6
    TABLEAU DES PROBABILITES
  (REP(O 6)INDICE LES COLONNES,
   REP(O 5)INDICE LES LIGNES)
 9   2   1   0   0   0   0   0   0   0   0   0   0   0   0   0
 1   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
 0   3   3   3   1   0   0   0   0   0   0   0   0   0   0   0
 0   1   1   4   1   1   0   0   0   0   0   0   0   0   0   0
 0   0   1   1   1   2   1   0   0   0   0   0   0   0   0   0
 0   0   3   1   2   0   0   0   0   0   0   0   0   0   0   0
 0   1   0   2   2   5   0   2   1   0   0   0   0   0   0   0
 0   0   0   0   2   4   6   1   1   1   0   0   0   0   0   0
 0   0   0   0   0   0   4   3   6   2   2   0   0   0   0   0
 0   0   0   0   3   1   2   3   2   5   7   5   1   0   0   0
 0   0   0   0   1   4   3   2   4   0  10   6   7   0   0   0
 0   0   0   0   0   3   5   8   4   1   4  11  13   6   1   0
 0   0   0   0   0   0   0   7   7   1   4   6   6   4   4   0
 0   0   0   0   0   0   0   2   5   5   1   4   2   2   2   0
 0   0   0   0   0   0   0   0   0   1   1   0   0   0   0   0
 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
                I( 5* 6;0M)=  1.9215
                H(0M)=  2.8409
 OPERATEUR A 255 REPONSES:INFO=  1.9215
 OPERATEUR A 254 REPONSES:INFO=  1.9215
 OPERATEUR A 253 REPONSES:INFO=  1.9215
 OPERATEUR A 252 REPONSES:INFO=  1.9215
 OPERATEUR A 251 REPONSES:INFO=  1.9215
 OPERATEUR A 250 REPONSES:INFO=  1.9215
 OPERATEUR A 249 REPONSES:INFO=  1.9215
 OPERATEUR A 248 REPONSES:INFO=  1.9215
 OPERATEUR A 247 REPONSES:INFO=  1.9215
 OPERATEUR A 246 REPONSES:INFO=  1.9215
 OPERATEUR A 245 REPONSES:INFO=  1.9215
 OPERATEUR A 244 REPONSES:INFO=  1.9215
 OPERATEUR A 243 REPONSES:INFO=  1.9215
 OPERATEUR A 242 REPONSES:INFO=  1.9215
 OPERATEUR A 241 REPONSES:INFO=  1.9215
 OPERATEUR A 240 REPONSES:INFO=  1.9215
 OPERATEUR A 239 REPONSES:INFO=  1.9215
 OPERATEUR A 238 REPONSES:INFO=  1.9215
 OPERATEUR A 237 REPONSES:INFO=  1.9215
 OPERATEUR A 236 REPONSES:INFO=  1.9215
 OPERATEUR A 235 REPONSES:INFO=  1.9215
 OPERATEUR A 234 REPONSES:INFO=  1.9215
 OPERATEUR A 233 REPONSES:INFO=  1.9215
 OPERATEUR A 232 REPONSES:INFO=  1.9215
 OPERATEUR A 231 REPONSES:INFO=  1.9215
 OPERATEUR A 230 REPONSES:INFO=  1.9215
 OPERATEUR A 229 REPONSES:INFO=  1.9215
 OPERATEUR A 228 REPONSES:INFO=  1.9215
 OPERATEUR A 227 REPONSES:INFO=  1.9215
*
```

Figure 28-a : Début du listing correspondant à un regroupent rapide

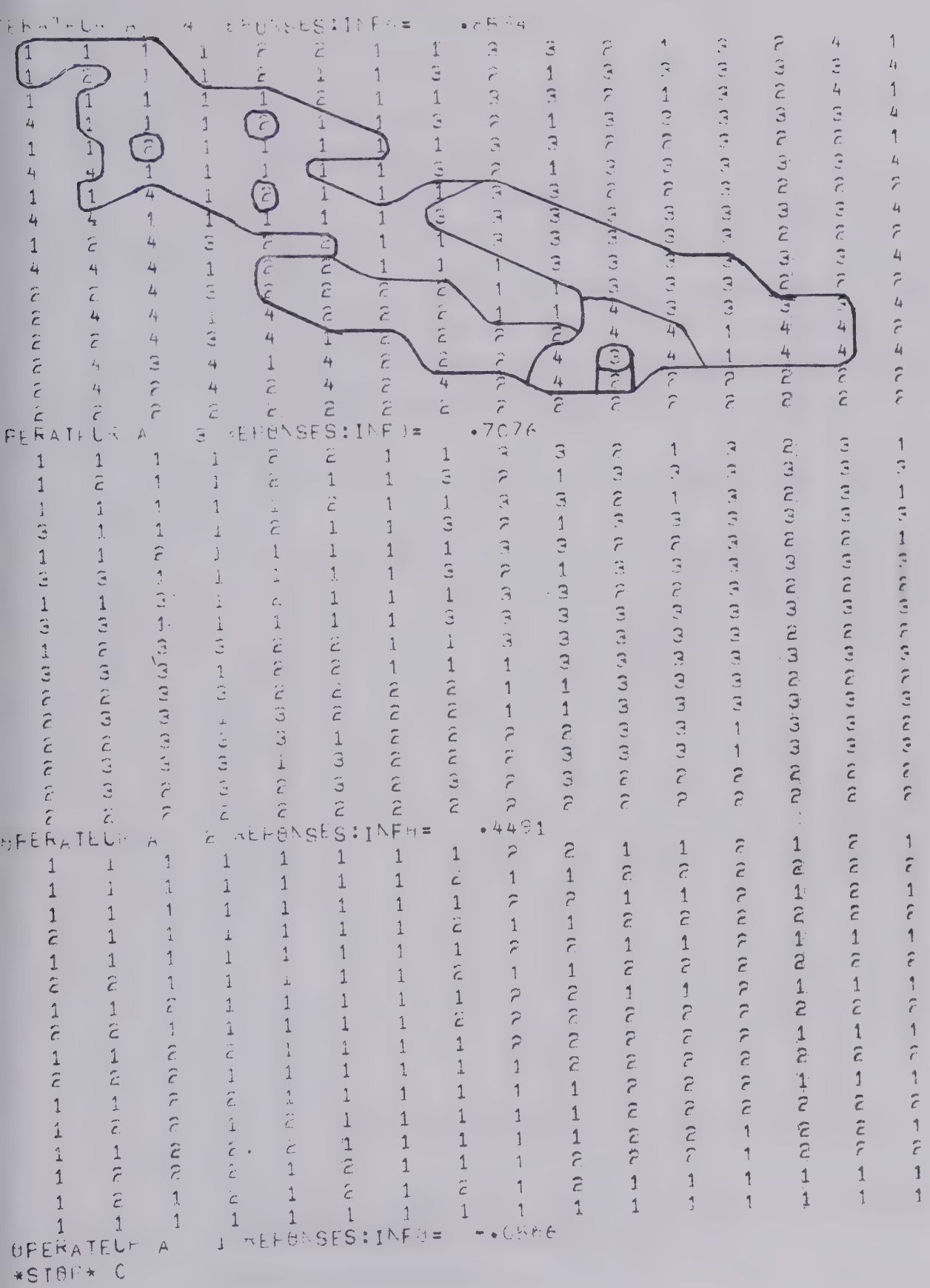


Figure 28-b : Fin du listing correspondant à un regroupement rapide

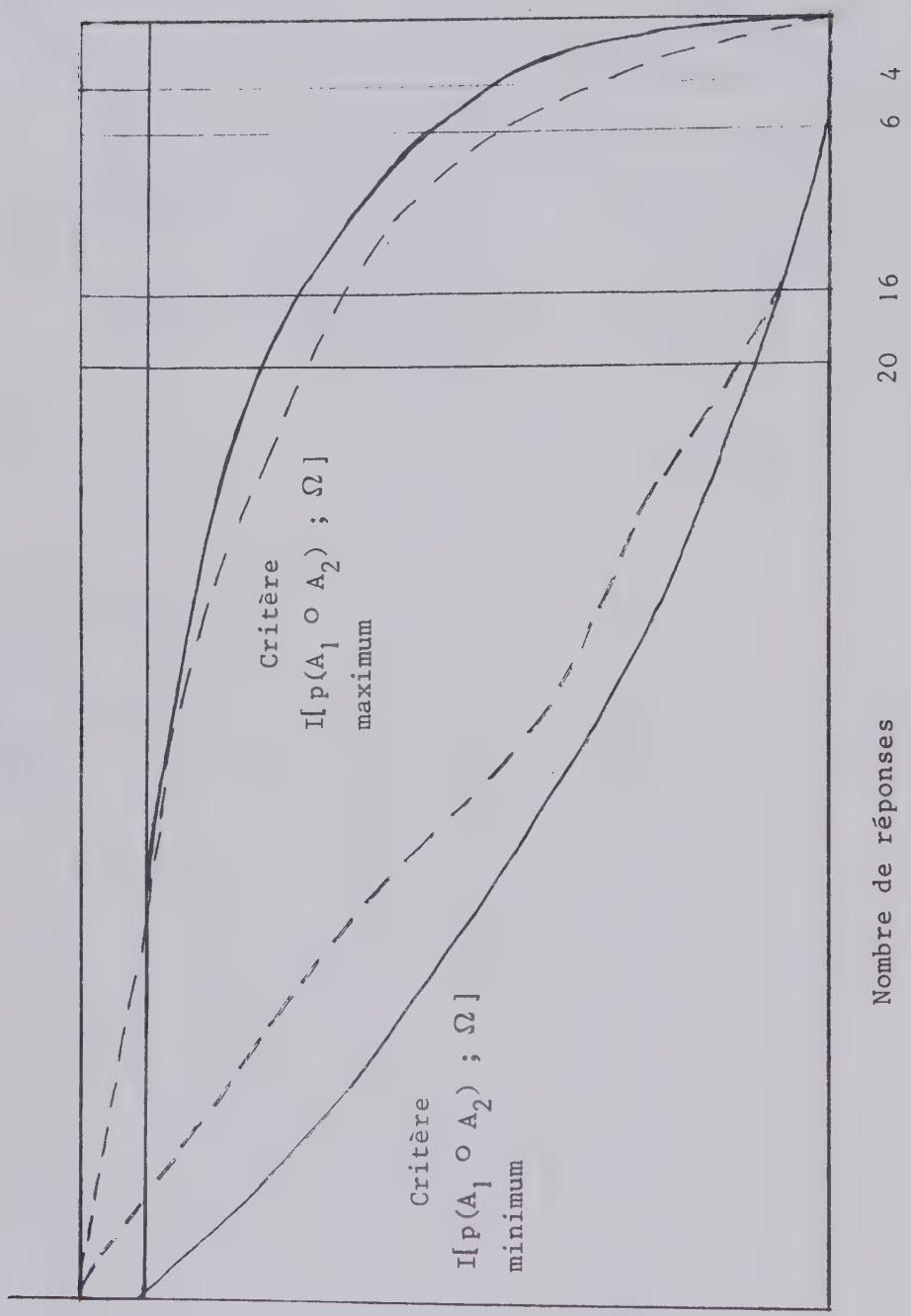


Figure 29 : Valeurs de $I[p(A_1 \circ A_2) ; \Omega]$ (en trait plein) et de $H[p(A_1 \circ A_2)]$ (en pointillé) suivant les critères $I[p(A_1 \circ A_2) ; \Omega]$ maximum et minimum

Les valeurs de l'information utile obtenues étaient très voisines et leurs différences inférieures à la précision du calcul de l'information mutuelle sur l'ordinateur.

Nous avons donc choisi dans nos expériences les m premières de la liste, ce qui permet d'accélérer le processus de convergence par des simplifications de programmation.

Les courbes donnant l'information mutuelle sont au début horizontales. Elles ne baissent rapidement que vers la fin de la convergence : pour 10 réponses l'information utile transmise n'est que 60 % de celle de $I[A_1, A_2 ; \Omega]$. Il faut 20 réponses pour que $I[p[A_1 \otimes A_2] ; \Omega]$ soit 77 % de $I[A_1, A_2 ; \Omega]$.

Plusieurs possibilités s'offrent à l'utilisateur pour la génération de l'opérateur :

Ou bien il veut un opérateur ne diminuant pas trop l'information utile : il dispose de peu de données, par ailleurs peu redondantes donc l'information qu'il perd ne pourra guère être retrouvée dans les autres données. Il impose alors la baisse d'information permise : la courbe donne donc le nombre de réponses de l'opérateur engendré.

Ou bien ses données offrent beaucoup de redondance et il faut d'abord réduire l'occupation de la mémoire, donc limiter les réponses de ses opérateurs, (le nombre de ces réponses influe directement sur la taille des tables des opérateurs qui utiliseront ces réponses comme données d'entrée). On impose alors le nombre des réponses : la courbe donne donc l'information utile apportée par l'opérateur créé.

C'est cette dernière possibilité que nous avons retenue dans notre application à la reconnaissance de phonèmes. Ces courbes que nous venons de présenter indiquent que si l'on veut avoir une bonne précision sur la valeur de l'optimum de l'information utile il faut choisir m supérieur au nombre de réponses désiré plus 6.

Des expériences faites, nous avons pu tirer un certain nombre de résultats :

Les deux courbes obtenues en utilisant cette méthode de regroupement avec les critères $I[A ; \Omega]$ maximum et minimum, déterminent une zone qui

représente, à la précision de la méthode près, le domaine des valeurs possibles de $I[A ; \Omega]$ en fonction du nombre de réponses de l'opérateur considéré. En particulier, pour des valeurs de ce nombre voisines de 16 (valeurs souvent utilisées), l'information mutuelle peut varier de valeurs presque nulles à des valeurs approchant $I[A_1 \otimes A_2 ; \Omega]$.

Il est donc très important de bien choisir l'opérateur utilisé.

Dans les exemples étudiés (résultats de capteurs acoustiques), les tableaux de regroupement montrent que les réponses qui se regroupent de préférence sont des réponses voisines géographiquement. Mieux : dans l'exemple précédemment exposé, les regroupements se font parallèlement à la diagonale de la table.

Ici l'opérateur $A_1 - A_2$ serait donc aussi très informant. On peut même remarquer que ce parallélisme à la diagonale est valable dans la zone où $A_1 \leqslant 11$. Dans le cas contraire les regroupements préférentiels se font dans le sens des colonnes. D'où l'opérateur :

Si $A_1 \leqslant 11$, alors $A = A_1 - A_2$

Sinon $A = A_1$

On conçoit donc ici comment ces méthodes peuvent aboutir à une génération d'opérateurs de reconnaissance, par la génération de la table et l'utilisation en conversationnel de cette table par un chercheur humain : l'opérateur complexe a été défini en ordinateur par sa table de correspondance, l'homme crée la structure correspondante au vu de cette table. Cette structure utilise dans ce cas des opérateurs simples tels que saut conditionnel, addition etc...

La méthode conversationnelle ainsi exposée est utilisable surtout lorsque l'occupation mémoire en ordinateur de la structure est inférieure à l'occupation de la table, ou bien que le cablage correspondant est d'un coût (rapidité et coût économique) plus faible que le cablage de la table.

Dans tout ce qui suivra nous n'utiliserons pas cette conception conversationnelle. NOus représenterons simplement tout opérateur par sa table.

VI.2.2. Etude de regroupements utilisant d'autres critères que l'information mutuelle

Nous avons voulu avec le même programme étudier les regroupements de réponses obtenues en prenant comme critère les critères dépendant de l'entropie, de l'information mutuelle avec un autre opérateur $O_3(I[p(A_1 \otimes A_2); A_3])$ et de la combinaison de cette information mutuelle avec $I[p(A_1 \otimes A_2) ; A_3]$). Les courbes présentées montrent les évolutions comparées de ces quantités obtenues à partir des données vocales déjà considérées (canaux 1 et 2 du vocoder).

VI.2.2.1. Critère $H[p[A_1 \otimes A_2]]$ maximum

L'algorithme obtenu en optimisant à chaque itération ce critère, est identique à l'algorithme de HUFFMANN. Il tend à créer des classes équiprobables à chaque itération. On constate sur les tableaux de regroupement à 16, 8 et 4 réponses que les classes sont géographiquement très dispersées (figure 31), contrairement à l'algorithme utilisant l'information mutuelle $I[p[A_1 \otimes A_2] ; \Omega]$.

Rappelons que la fonction $H[p[A_1 \otimes A_2]]$ a été définie comme représentant l'information totale apportée par l'opérateur. Elle est différente de $I[p[A_1 \otimes A_2] ; \Omega]$ qui représente l'information utile.

La différence des deux, qui représente l'"information inutile", est $H[p[A_1 \otimes A_2] \mid \Omega]$;

Il est intéressant de constater que l'information inutile reste pratiquement constante jusqu'au regroupement correspondant à un opérateur ayant 25 réponses (partition à 25 parties) contrairement à l'expérience de la figure 29.

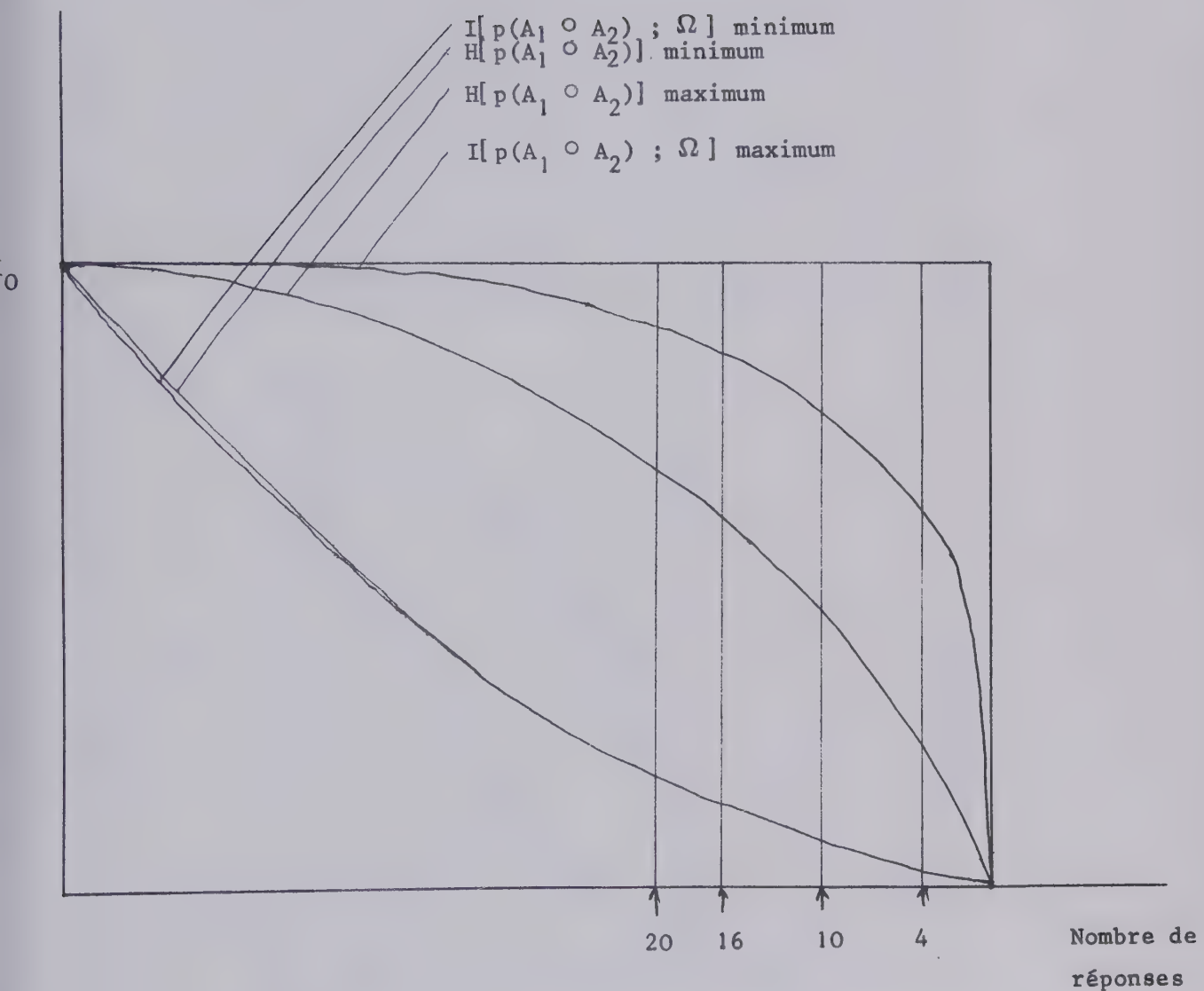


Figure 30 : Variations de $I[p(A_1 \circ A_2) ; \Omega]$ avec divers critères de regroupements

Regroupement avec $H[p(A_1 \times A_2)]$ max

16 réponses

O_2 \ O_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1																
2																
3				11	8											
4			12	12	12											
5					1	9	12	12	12	11	11					
6				13	16		8	2	16	16	2	10				
7					10	13	13	-10	2	12	5	1	1			
8						12	12	8	10	8	2	4	15			
9									5	10	3	3	11	13		
10								10	3	13	12	1	3	10		
11										7	11	1	15	9		
12										9	7	6	14	1	8	
13											6	4	9	8		
14											8	14	5	8		
15												16	8	16		
16																

Regroupements suivants :

15-16 ⌊ 10-14 ⌊ 11-12 ⌊ 9-13 ⌊ 5-7 ⌊ 6-8

Figure 31-a : Valeurs de $H[p(A_1 \circ A_2)]$ en fonction de divers regroupements

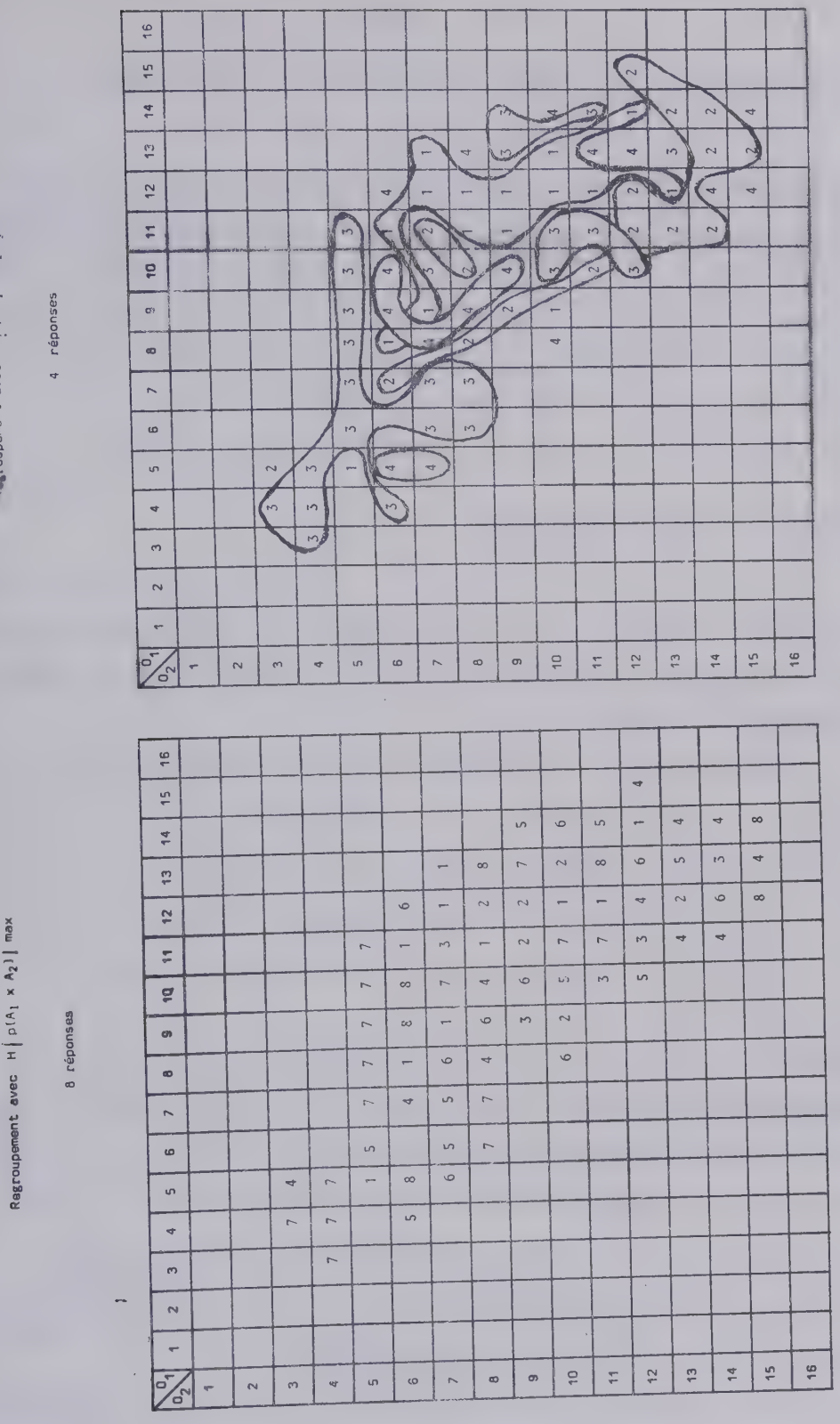


Figure 31-b : Regroupement des réponses avec comme critère $H[p(A_1 \circ A_2)]$ maximum

VI.2.2.2. Critère $H[p(A_1 \otimes A_2)]$ minimum

Ce critère symétrique du précédent montre un regroupement en tache d'huile : pour une partition de 16 classes, 15 des 16 classes sont formées chacune d'une seule réponse élémentaire et la 16ème du regroupement de toutes les autres réponses. La perte d'information utile est très rapide. A l'itération correspondant à l'opérateur ayant 25 réponses, l'information utile $I[p(A_1 \otimes A_2) ; \Omega]$ n'est plus que le 1/5ème de $I[A_1 \otimes A_2 ; \Omega]$.

VI.2.2.3. Remarques sur ces deux critères

Nous avions voulu étudier dans quelle mesure les regroupements suivant l'information totale maximum étaient correlés avec ceux qui correspondent à la maximisation de l'information utile.

Dans les exemples étudiés, on constate que les courbes représentatives ont un comportement identique. En particulier, les figures 29 et 31 montrent les courbes représentatives de $H[p(A_1 \otimes A_2)]$ lorsqu'on effectuait le regroupement des réponses $(A_1 \otimes A_2)$ suivant le critère $I[p[A_1 \otimes A_2] ; \Omega]$ maximum. Ces courbes sont données aussi page 141. On remarque sur toutes ces courbes bes que les deux quantités information totale et information utile ont des variations correlées.

Cette corrélation permettrait elle d'envisager les possibilités d'un apprentissage sans professeur ? Les études sont en cours dans ce sens. Ce fait permettrait donc de définir une forme indépendamment de tout contexte humain et de tout opérateur O_{id}, comme il est fait tout au long de cette étude.

La fonction $H[A]$ mesure la manière dont s'effectue l'occupation mémoire des réponses A de l'opérateur O et son optimum est atteint lorsque toutes les réponses A sont utilisées équiprobablement.

Une forme correspondrait alors aux résultats d'opérateurs O implantés de telle manière que l'occupation de la mémoire par les données de l'ensemble d'apprentissage soit optimal.

VI.2.2.4. Critère $I[p(A_1 \circledast A_2) ; A_3]$ minimum

Nous avons vu dans les premiers chapitres qu'il était important lorsque deux variables A_i et A_j doivent être associées en vue de reconnaître les formes Ω_r que ces deux variables soient le moins correlées possible.

En effet, la relation :

$$I[A_1, A_2 ; \Omega] \geqslant I[A_1 ; \Omega] + I[A_2 ; \Omega] - I[A_1 ; A_2]$$

donne une borne inférieure à l'information utile apportée par l'ensemble de A_1 et A_2 ; cette borne dépend de la valeur de $I[A_1 ; A_2]$ qui mesure la corrélation entre les variables A_1 et A_2.

En effet, $I[A_1 ; A_2]$ est l'espérance mathématique sur l'ensemble des différents $a_1(k_1)$ et $a_2(k_2)$ du logarithme de

$$\frac{p[a_1(k_1), a_2(k_2)]}{p[a_1(k_2)] \cdot p[a_2(k_2)]}$$

qui est le coefficient de corrélation entre les réponses $a_1(k_1)$ et $a_2(k_2)$.

Soient 3 variables A_1, A_2, A_3 ; si nous créons un opérateur travaillant sur les données A_1 et A_2 déterminant la valeur de la variable $p(A_1 \circledast A_2)$, il est intéressant d'étudier dans quelle mesure on peut le créer de telle façon que la fonction $I[p(A_1 \circledast A_2) ; A_3]$ soit la plus petite possible.

On a appliqué l'algorithme de regroupement précédemment décrit sur les variables A_1 et A_2 en prenant comme critère de regroupement la minimisation de la fonction $I[p(A_1 \circledast A_2) ; A_3]$ (données vocales : les variables A_1 A_2 et A_3 sont les résultats des 3 premiers filtres du vocoder).

Quand l'opérateur créé a 45 réponses, $I[p(A_1 \circledast A_2) ; A_3]$ a diminué de 85 % par rapport à sa valeur initiale et $H[p(A_1 \circledast A_2)]$ seulement de 40 %.

Pour 20 réponses de l'opérateur, $I[p(A_1 \circledast A_2) ; A_3]$ a diminué de 95 % et $H[p(A_1 \circledast A_2)]$ de 60 %.

A ce moment $p(A_1 \circledast A_2)$ est pratiquement indépendante de A_3 et reste quand même informant.

La variable créée est a fortiori indépendante des 9 autres opérateurs canaux puisque c'est le résultat A_3 du 3ème filtre qui est le plus corrélé avec A_1 et A_2.

Le tableau des regroupements montre des classes très dispersées.

VI.2.2.5. Critère $I[p(A_1 \circledast A_2) ; A_3]$ maximum

Le critère inverse du précédent, conduit à un algorithme qui donne la corrélation avec A_3 de l'ensemble des variables A_1 et l_2. Plus exactement, on a en fonction de n la valeur maximum de la corrélation $I[A ; A_3]$ avec O_3 d'un opérateur à n réponses travaillant sur A_1 et A_2. Voir figure 34 : $I[p(A_1 \circledast A_2) ; A_3]$ décroit très lentement jusqu'au 85ème regroupement (20 réponses pour l'opérateur créé).

Dans ce cas, les classes obtenues sont des ensembles connexes approximativement parallèle à l'axe correspondant aux valeurs de A_1. On peut expliquer ce fait, par ce que A_2 est plus correlée avec A_3 que ne l'est A_1.

On pourrait approcher l'opérateur obtenu par une combinaison linéaire simple de A_1 et A_2.

VI.2.2.6. Critère $I[p(A_1 \circledast A_2) ; \Omega] - I[p(A_1 \circledast A_2) ; A_3]$ maximum

Ce critère est tiré directement de l'inégalité donnant une borne inférieure à $I[A_1, A_2 ; \Omega]$. C'est d'après cette inégalité, le critère qui donne a priori le maximum de l'espérance mathématique de la valeur de :

$$I[p(A_1 \circledast A_2), A_3 ; \Omega]$$

Figure 32-a : Critère $I[p(A_1 \circ A_2) ; A_3]$ maximum

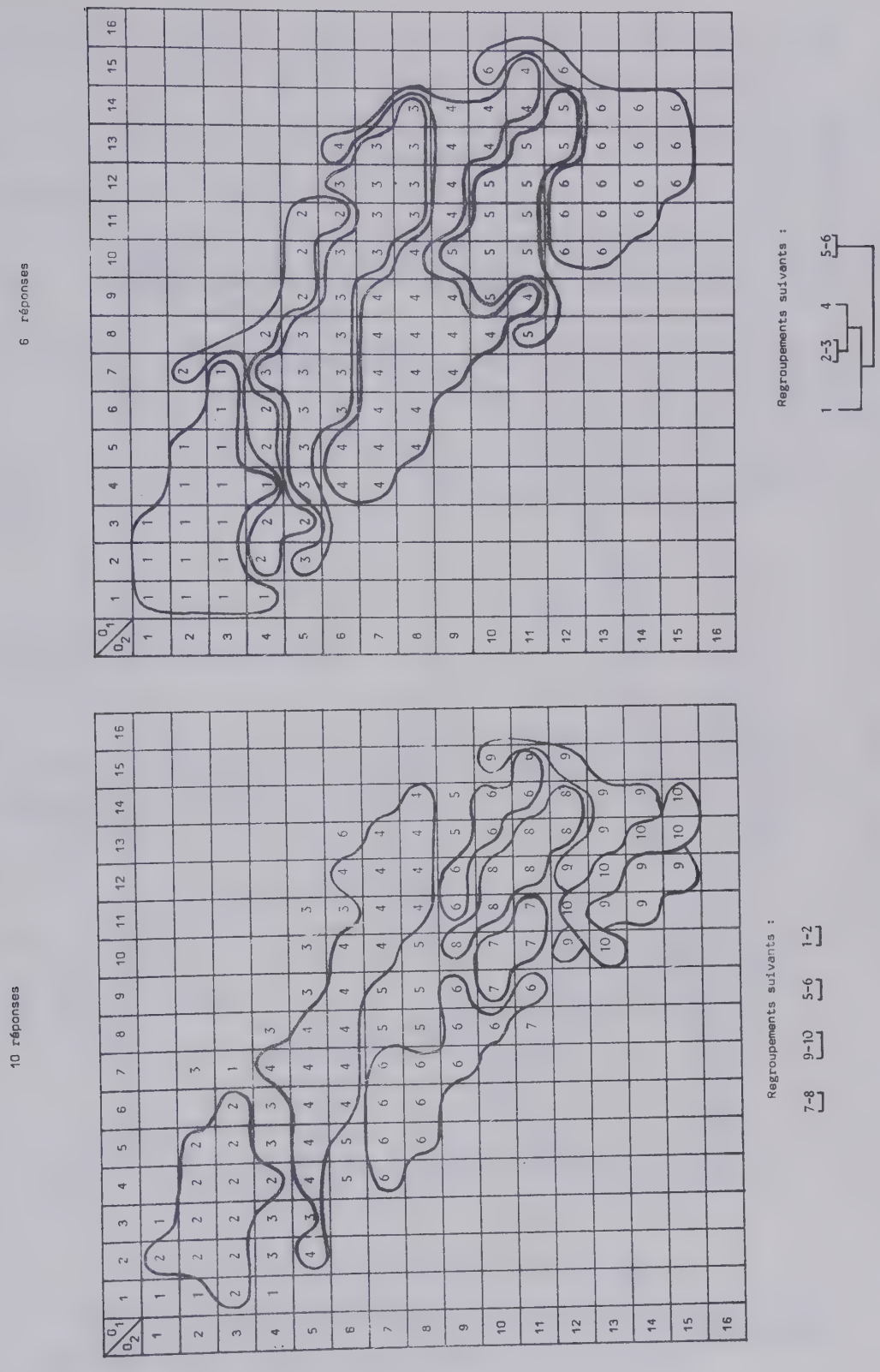


Figure 32-b : Critère $I[p(A_1 \circ A_2) ; A_3]$ maximum

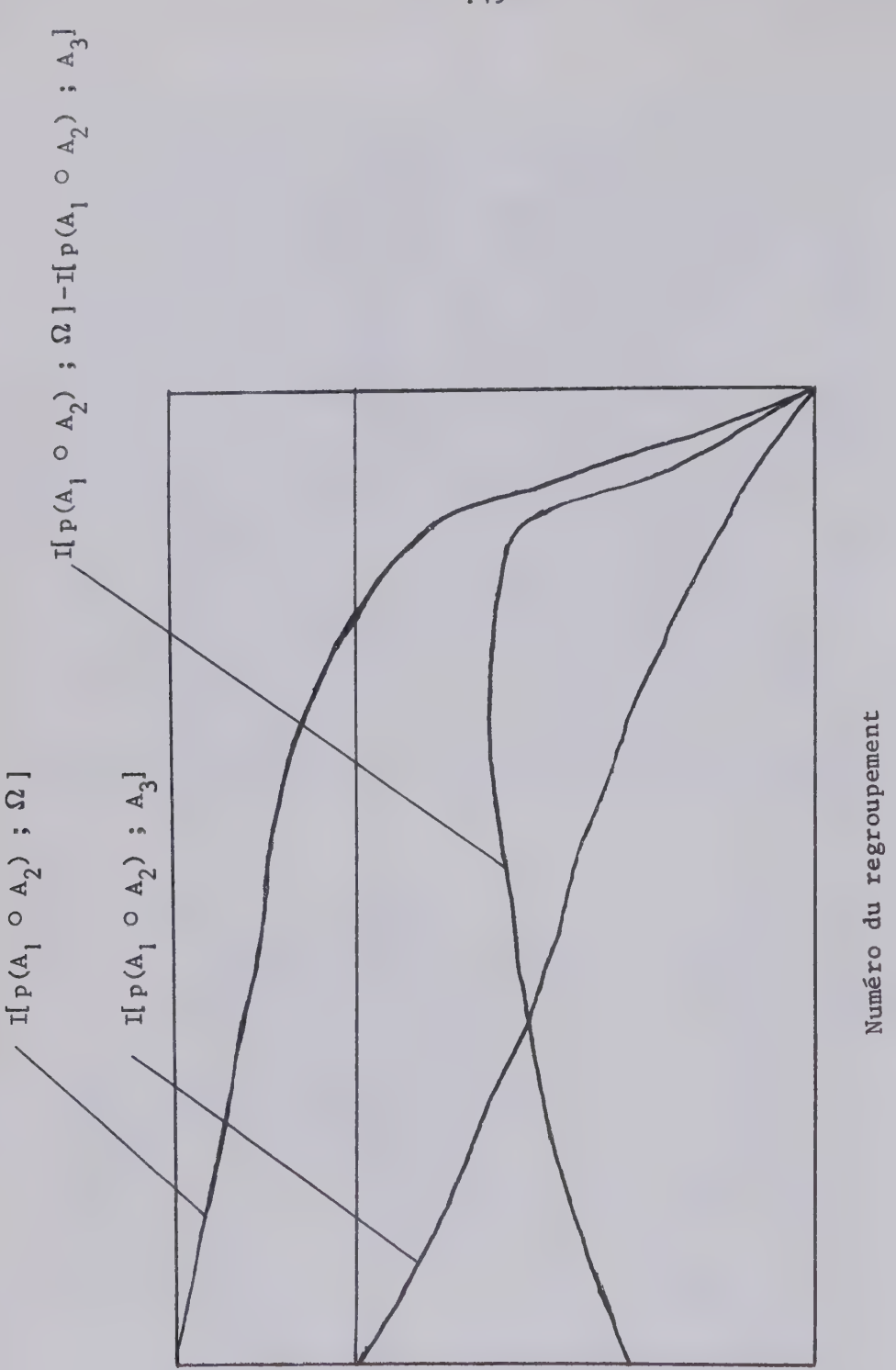


Figure 33 : Regroupement des réponses suivant le critère $I[p(A_1 \circ A_2) ; \Omega] - I[p(A_1 \circ A_2) ; A_3]$ maximum

On montre sur la figure 33 les variations de $I[p(A_1 \circledast A_2) ; \Omega]$ de $I[p(A_1 \circledast A_2) ; A_3]$ et de la différence entre ces deux valeurs.

On remarque que cette différence présente un maximum qui correspond ici à une partition $p(A_1 \circledast A_2)$ de 22 parties (opérateur à 22 réponses). Pour ce maximum $I[p(A_1 \circledast A_2) ; \Omega]$ a diminué de 23 % par rapport à $I[A_1, A_2 ; \Omega]$ et $I[p(A_1 \circledast A_2) ; A_3]$ de 62 % par rapport à $I[A_1, A_2 ; A_3]$. L'opérateur construit reste donc très informant en étant peu correlé avec les autres.

Le tableau des regroupements montre des classes très dispersées, l'opérateur créé ne peut donc s'approcher par une suite d'opérations simples à partir de A_1 et A_2 comme pour l'opérateur du paragraphe VI.2.1. Il serait donc nécessaire de le définir par la table de son opération associée si on voulait l'utiliser.

VI.3. GENERATION DE LA STRUCTURE

VI.3.1. Principe et description d'un programme général de création de structures de reconnaissance

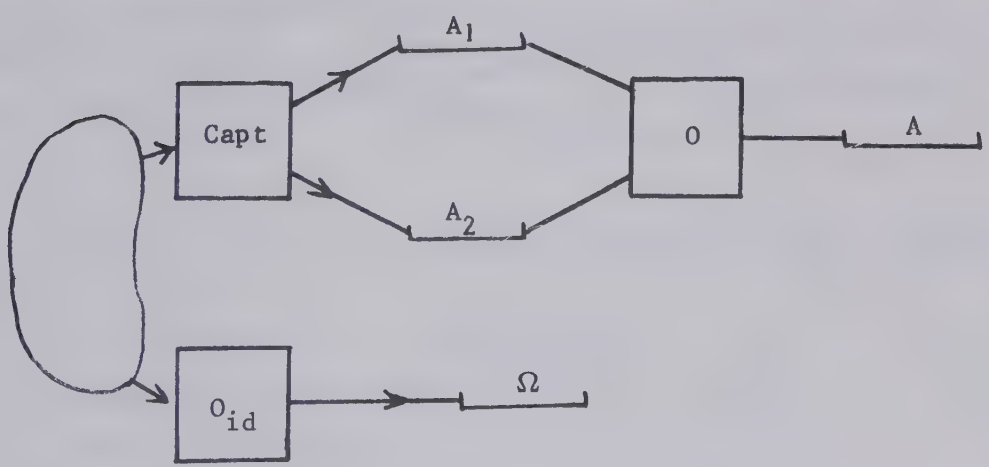


Figure 34

La figure 34 symbolise un problème de reconnaissance de formes à 2 variables. Un (ou plusieurs) appareil de mesure induit dans chacune des variables A_1 et A_2 un certain état ($a_1(k_1)$ et $a_2(k_2)$).

Un autre opérateur O_{id} (l'homme par exemple) fait de même dans la variable Ω. Les états de Ω sont les formes que "représentent" peut être les variables A_1 et A_2.

Le problème de la reconnaissance des formes Ω à partir des données $a_1(k_1)$ et $a_2(k_2)$, est de trouver un opérateur O dont A_1 et A_2 sont les variables d'entrée et A la variable de sortie tel que

$$I[\Omega ; A]$$

soit le plus proche possible de $H[\Omega]$.

Si nous définissons l'opération de O par sa table, la méthode du paragraphe précédent résoud le problème : on arrête l'algorithme de regroupement lorsque le nombre d'états distincts de A est égal au nombre de formes $card(\Omega)$.

La figure 35 par contre symbolise un problème de reconnaissance de formes à plus de 2 variables.

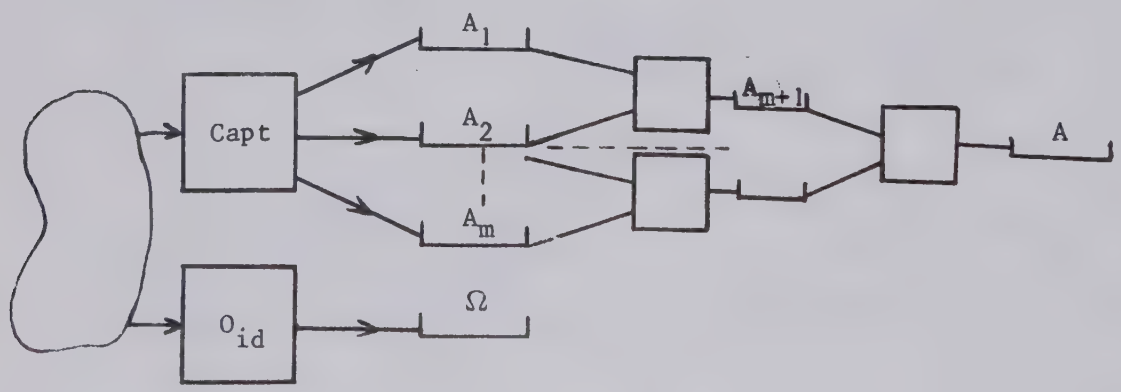


Figure 35

Le but est de même de disposer d'une variable A telle que

$$I[\Omega ; A]$$

soit le plus proche possible de $H[\Omega]$.

Si nous ne voulons utiliser que des opérateurs à deux variables d'entrée et une variable de sortie, il devient nécessaire de construire une structure de tels opérateurs (chapitre I).

Les hypothèses que nous avons faites pour la construction de cette structure dans tous les problèmes pratiques que nous avons étudiés sont les suivantes :

1. $I[\Omega ; A]$ est le plus voisin possible de $H[\Omega]$ sur un ensemble de données d'apprentissage a priori.
2. Le graphe de la structure est une arborescence.
3. Les opérateurs de la structure ont deux variables d'entrée et une variable de sortie.
4. Leurs opérations sont définies par leur table.
5. Le nombre d'états de A est égal à $card[\Omega]$.
6. Le nombre d'états des variables intermédiaires est le même pour toutes ces variables. C'est le paramètre n_Ω de l'algorithme.

Les hypothèses 2, 3 et 6 sont restrictives. Par contre l'hypothèse 4 enlève la restriction d'usage dans les méthodes classiques de n'utiliser que les opérations qui portent un nom comme l'addition, la multiplication etc...

L'algorithme que nous présentons construit cette structure de la manière suivante :

A chaque itération j un opérateur O_j est créé. Si m est le nombre de variables d'entrée, les hypothèses 2 et 3 impliquent (m - 1) itérations.

Soit $\mathcal{L}_j$ l'ensemble des variables dites "utilisables" à l'itération j : deux des éléments de $\mathcal{L}_j$ sont choisis et constituent les variables d'entrée de l'opération O_j.

Ce choix est fait en prenant les deux variables A_{j_1} et A_{j_2} correspondant au minimum de

$$I[\Omega ; A_{j_1}] + I[\Omega ; A_{j_2}] - I[A_{j_1} ; A_{j_2}]$$

La table de l'opération de O_j est créée par la méthode décrite précédemment : on cherche à chaque itération la table de l'opérateur O_j qui transmette le plus d'information utile possible. Pour $j < m - 1$, le paramètre n_r de l'algorithme fixe le nombre de réponses différentes de l'opérateur O_j. Si par contre $j = m - 1$, ce nombre de réponses est égal à card(Ω).

La variable de sortie de O_j, A_{m+j} est ajoutée à L_j, les deux variables A_{j_1} et A_{j_2} sont retirées. Nous obtenons ainsi le nouvel ensemble L_{j+1} des variables utilisables et nous itérons le procédé.

L'ensemble L_1 est l'ensemble des variables d'entrée de la structure à engendrer (A_1, ..., A_m). L'ensemble L_{m-1} est constitué de la seule variable A_{2m-1} qui est la variable A de sortie de la structure engendrée.

L'algorithme présenté ajoute donc les deux hypothèses restrictives suivantes :

7. On cherche à optimiser à chaque itération la fonction $I[\Omega ; A_{m+j}]$.
8. Le choix des variables d'entrée des opérateurs O_j se fait suivant les valeurs de

$$I[A_{j_1} ; \Omega] + I[A_{j_2} ; \Omega] - I[A_{j_1} ; A_{j_2}] \text{ minimum}$$

L'hypothèse 7 est justifiée par le fait expérimental qu'un opérateur informant a en général des variables d'entrée informantes (inégalités sur les informations utile d'opérateurs série et parallèle).

Le choix des variables de l'hypothèse 8, ressemble à celui de l'algorithme de HUFFMANN dont le but est de construire le questionnaire le plus court ayant une information totale donnée. En effet, l'algorithme de HUFFMANN associe les états de variable dont les probabilités sont les plus faibles . Nous associons ici les variables A_{j_1} et A_{j_2} dont l'espérance a priori de

$$I[A_{j_1}, A_{j_2} ; \Omega]$$

est la plus faible.

A priori en effet nous ne connaissons de

$$I[A_{j_1}, A_{j_2} ; \Omega]$$

que sa borne inférieure : $I[A_{j_1} ; \Omega] + I[A_{j_2} ; \Omega] - I[A_{j_1} ; A_{j_2}]$.

Par ailleurs, il est plus simple (occupation mémoire et temps de calcul en ordinateur) de calculer cette dernière valeur que celle de $I[A_{j_1}, A_{j_2} ; \Omega]$

Ces hypothèses ont été introduites en vue d'accélérer l'algorithme d'apprentissage et de le rendre utilisable. Ces hypothèses permettent une création algorithmique sans processus d'essai aléatoire et de retour en arrière (back track), en choisissant à chaque alternative la branche qui a la plus grande probabilité a priori d'être la meilleure.

Elles ne sont encore l'objet d'aucun théorème et ne trouvent actuellement leur justification que par le fait qu'elles permettent par apprentissage la résolution de beaucoup de problèmes pratiques qui n'avaient pas été résolus jusqu'à présent de manière satisfaisante par les méthodes classiques (paragraphes suivants).

VI.3.2. Introduction des coûts associés aux formes

Nous verrons dans un des problèmes étudiés à l'aide de cet algorithme qu'il a été nécessaire d'introduire des coûts associés aux formes à reconnaître.

Ces coûts représentent en fait des importances affectées à chacune des formes. Voir [57]. Ce sont des nombres entiers :

$$\{c(i) \mid i = 1 \text{ à } n\}$$

Dans les calculs de toutes les fonctions $I[A_i ; A_j]$, $H[\Omega]$ et $I[\Omega ; A_j]$, nous remplaçons les fréquences

$n[\omega(i), a_j(k)]$ par $c(i) \cdot n[\omega(i), a_j(k)] = n'[\omega(i), a_k(k)]$

$n[\omega(i)]$ par $c(i) \cdot n[\omega(i)] = n'[\omega(i)]$

$n[a_j(k)]$ par $\sum_i c(i) \cdot n[\omega(i), a_j(k)] = n'[a_j(k)]$

Pour le calcul des probabilités, au lieu de

$$N = \sum_{i,k} n[\omega(i), a_j(k)]$$

nous calculons :

$$N' = \sum_{i,k} n'[\omega(i), a_j(k)]$$

Au lieu d'assimiler les probabilités $p[\omega(i), a_j(k)]$ à $\frac{1}{N} n(\omega(i), a_j(k)]$ nous utilisons les nombres :

$$p'[\omega(i), a_j(k)] = \frac{n'[\omega(i), a_j(k)]}{N'}$$

On vérifie aisément que lorsque N tend vers l'infini, les limites de ces valeurs p', vérifient les relations caractéristiques des probabilités.

De même, les fonctions $H'[\Omega]$, $H'[A_j]$, $I'[\Omega ; A_j]$ calculées sur ces nouvelles valeurs vérifient les mêmes relations que celles décrites au chapitre III.

Le paragraphe VI.5. décrit une utilisation de ces coûts.

VI.3.3. L'étude de l'ergodicité

Un problème important, ici de même qu'aux chapitres précédents, est celui de la mesure des probabilités utilisées pour le calcul de $I[A ; \Omega]$:

$$p[a(k) ; \omega(i)]$$

et surtout le problème de leur validité.

La mesure de ces probabilités a été faite par la mesure des fréquences d'apparitions $n[a(k) ; \omega(i)]$ sur un ensemble de données appelé ensemble d'apprentissage.

1. Il faut que ces fréquences soient suffisamment grandes pour représenter les probabilités à la précision voulu (dans le cas d'un ensemble ergodique de données.

Appelons N le nombre d'expériences, n le nombre de formes et n_r le paramètre de l'algorithme fixant le nombre de réponses de chaque opérateur. La valeur moyenne des fréquences $n[a(k) ; \omega(i)]$ vaut : $\frac{N}{n \cdot n_r}$.

La valeur de n ne peut être modifiée. Par contre n_r est laissée au choix de l'expérimentateur. Il faut évidemment avoir le nombre N d'expériences le plus grand possible. La valeur de n_r s'en déduit en prenant le maximum possible compte tenu de la précision voulue sur la mesure des probabilités.

2. Nous mesurons cette précision en considérant comme aux chapitres précédents, deux ensembles : un ensemble d'apprentissage et un ensemble de test. Les fréquences sont calculées sur l'ensemble d'apprentissage. L'algorithme d'apprentissage travaille sur cet ensemble. Une fois la structure de reconnaissance créée, on teste finalement celle-ci sur l'ensemble de test. Les performances définitives de la structure sont celles qui sont mesurées dans ce dernier ensemble.

L'ergodicité de l'ensemble de données et la validité des mesures de probabilité sont mises en évidence en comparant les tables de confusion de la structure de reconnaissance d'une part sur l'ensemble d'apprentissage, d'autre part sur l'ensemble de test.

On peut donc au vu de ces deux tables de confusion déterminer la valeur optimale du paramètre n_r. Il y a un compromis à déterminer entre trois facteurs :

- la précision voulue sur la mesure des probabilités
- l'occupation mémoire de la structure de reconnaissance.

En effet, si les variables intermédiaires ont chacune n_r états, l'occupation mémoire des tables des opérateurs ayant celles-ci comme variables d'entrée, est proportionnelle à n_r^2.

- la richesse de la structure qui varie en même temps que n_r et dont dépendent directement les performances finales.

VI.3.4. Programmation de l'algorithme d'apprentissage et ses performances

L'algorithme d'apprentissage a été programmé en FORTRAN. Ses caractéristiques sont les suivantes :

- 630 instructions FORTRAN
- occupation mémoire : 12 100 mots de 24 bits
- ordinateur utilisé CII 10 070
- paramètres à déterminer :

NBEXP : nombre d'expériences de l'ensemble d'apprentissage. Valeur entière inférieure à 2 000.

NBFO : nombre de formes (n) jusqu'à 32

NBREP : paramètre n_r fixant le nombre d'états des variables intermédiaires : $n_r \leqslant 16$

NBDON : nombre de variables d'entrée de la structure à engendrer : NBDON $\leqslant$ 16

COUT(I),I = 1 à NBFO : coûts respectifs de chacune des formes à reconnaître.

- occupation mémoire de la structure engendrée :

  (NBFO) . $\frac{NBREP^2}{8}$ mots de 24 bits environ (au maximum donc 1K mots de 24 bits environ).

- temps d'exécution de l'algorithme d'apprentissage : dans les problèmes étudiés jusqu'à présent : de 8 à 18 mn y compris le temps de compilation (CII 10 070).

- temps d'exécution de l'algorithme de reconnaissance correspondant à la structure engendrée :

$$20 \times NBFO \times t \quad \text{environ}$$

(si t est le temps moyen d'exécution d'une instruction de l'ordinateur). Donc au maximum avec l'ordinateur CII 10 070 : 3 ms et au minimum 0,2 ms.

Après initialisation,

Le programme lit les cartes de données dont les NBEXP premières seront considérées comme ensemble d'apprentissage et toutes les suivantes comme ensemble de test.

Outre un certain nombre de résultats (valeurs de probabilités, tables des opérations créées, valeurs des fonctions $I[A_i ; A_j]$ et $I[A_i ; \Omega]$), le programme donne les deux tables de confusion, de la structure de reconnaissance sur les ensembles d'apprentissage et de test. Il donne aussi les cartes perforées correspondant à la structure de reconnaissance elle-même pour qu'elle puisse être utilisée de manière opérationnelle ultérieurement.

VI. 4. PROBLEME DE LA RECONNAISSANCE DE PHONEMES (DONNEES VOCALES)

Nous avons mis au point le programme d'apprentissage précédent sur les données vocales présentées aux paragraphes III.8.2. et III.8.3.

Les expériences dont nous disposions étaient en nombre très faible : 275 seulement. Nous n'avons pas partagé ces données en un ensemble d'apprentissage et un ensemble de test. Les performances obtenues n'ont ainsi que peu de sens : nous obtenons 3 % d'erreur seulement sur la reconnaissance des 24 phonèmes que nous nous proposions de reconnaître dans l'ensemble d'apprentissage lui-même. Ce taux d'erreur est trop faible pour ce que nous savons expérimentalement des performances du vocodeur : confusion entre les s et z, entre les m et n... Mais le programme d'apprentissage permettra sans doute (avec des données en plus grand nombre) d'arriver à extraire des données du vocodeur, un pourcentage important de l'information utile qu'il transmet.

Par contre, si les performances de la structure obtenue n'ont pas été testées de manière satisfaisante, il est intéressant de montrer la manière dont s'est construit le graphe de cette structure. En effet à chaque itération de l'algorithme d'apprentissage, les deux variables choisies pour servir d'entrée à l'opérateur créé sont voisines géographiquement :

considérons les 48 variables élémentaires constituées par les valeurs de sortie des 12 filtres du vocodeur à 4 instants successifs. L'information utile de chacune de ces variables dans le but de reconnaître les 24 phonèmes est donnée sur la table de la page 75 . Le graphe de la structure de reconnaissance créée peut être représenté par une hiérarchie sur ces mêmes variables. Le tableau suivant figure cette hiérarchie :

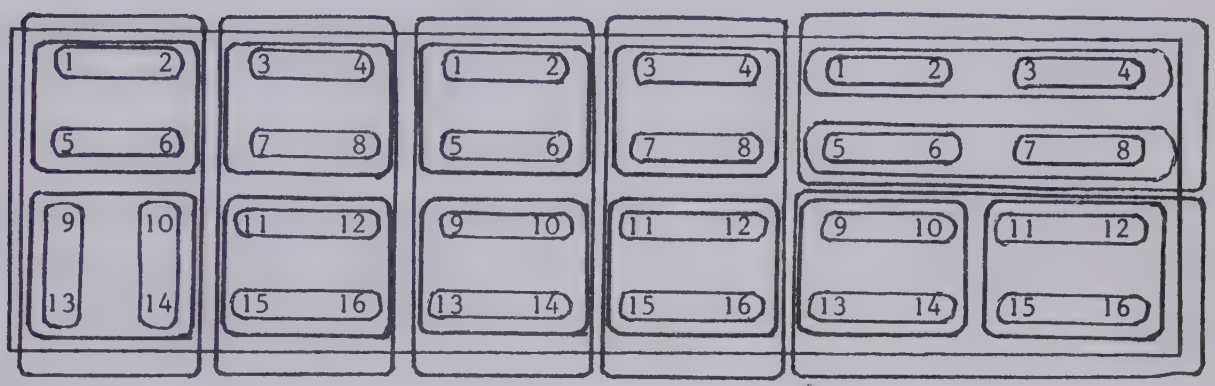


Figure 36

(Le critère de regroupement est le critère général du minimum de :

$$I[\Omega ; A_i] + I[\Omega ; A_j] - I[A_i ; A_j]$$

L'explication de ces regroupements privilégiés réside sans doute dans l'influence du terme $I[A_i ; A_j]$ qui est fonction décroissante de la distance entre A_i et A_j dans le plan temps-fréquence.

VI.5. PROBLEME DE LA SEGMENTATION DE LA PAROLE (DONNEES VOCALES)

Ce problème a été étudié avec la collaboration de l'équipe d'acousticiens du Laboratoire de Mécanique d'Orsay (LIENARD, MLOUKA).

Le but de cette équipe est de trouver les opérateurs de reconnaissance de phonatomes (ou diphonèmes) [43]. Ces opérateurs supposent une standardisation de leurs données vocales. D'où la nécessité d'indiquer sur ces données les frontières entre phonatomes. Ces frontières correspondent à peu près aux positions des phonèmes (en fonction du temps). Le paragraphe III.8.4. précédent montre les résultats des mesures de l'information utile des filtres en fonction de leur fréquence.

Les données vocales sont constituées de phrases prononcées par 8 locuteurs (4 hommes et 4 femmes), le tout représente 30 secondes de parole. Sur ces données, l'acousticien nous a indiqué la position des "top" de segmentation définissant les segments caractéristiques des diphonèmes à reconnaître.

Notre but est donc de trouver un opérateur travaillant sur les données et indiquant les mêmes positions pour ces "top" de segmentation. En se ramenant à un problème de reconnaissance de formes nous avons à créer sur ces données un opérateur qui reconnaît les formes "top" et "non top".
Voir ROCHE [65].

Les variables de base sont au nombre de 14. Chacune d'elles représente la somme de données situées dans un rectangle tel que sur la figure 37.

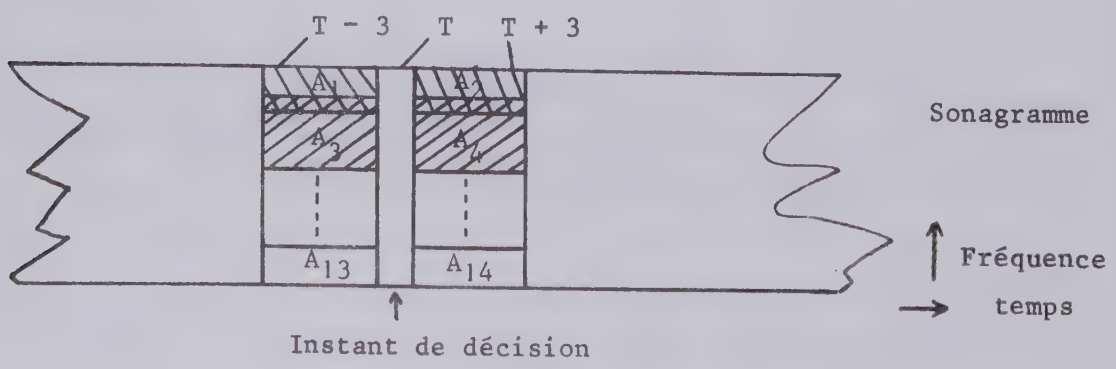


Figure 37

Les rectangles se recouvrent mutuellement en partie.

A ces 14 variables, on rajoute une 15ème qui tiendra compte du contexte grâce à une programmation dynamique : elle représente la durée qui s'est écoulée depuis le dernier top de segmentation détecté.

La notion d'importance relative affectée à chacune des formes a été introduite dans le programme général d'apprentissage pour la résolution de ce problème de segmentation. En effet, il est plus grave de ne pas connaître un top de segmentation que de décider d'un top aux instants où il ne devrait pas y en avoir.

Les performances de l'opérateur de segmentation engendré ont été les suivantes : (les paramètres utilisés sont : importance relative entre top non reconnu et top en trop = 6 ; occupation mémoire de la structure = 3 500 bits).

Sur les données d'apprentissage (82 top) :

79 top reconnus
3 top non reconnus
52 top en trop.

Sur les données de test (indépendantes des données d'apprentissage et introduites a posteriori : 57 top) :

45 top reconnus
12 top non reconnus
18 top en trop

Le temps mis pour l'apprentissage a été de 8 mn de CII 10 070, l'algorithme de segmentation engendré utilise 100 μs toutes les 10 ms (temps d'acquisition des données vocales).

La figure 38 montre un exemple de résultats (données de test) avec les deux sortes d'erreurs (top en trop, top non reconnu) : la première colonne de top correspond à la décision du spécialiste acousticien (LIENARD), la deuxième colonne à la décision de l'algorithme.

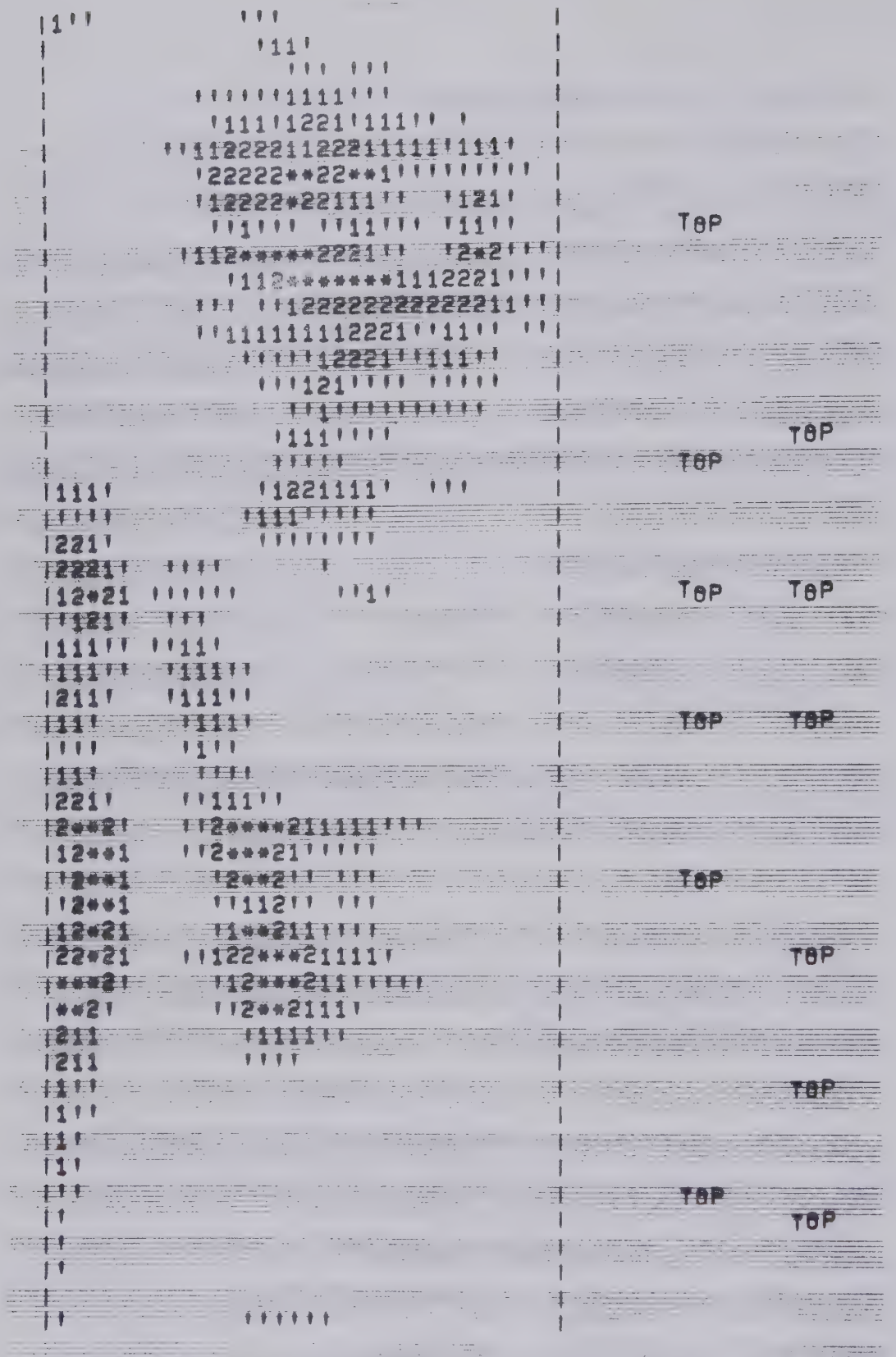


Figure 38

Remarque 1

La présence de la 15ème variable qui représente la durée écoulée depuis le dernier top, vient de la remarque suivante : si à un instant T on détecte un top, on peut en détecter aussi aux instants T + 1 et T - 1 sans qu'on les considère comme des erreurs ainsi qu'aux instants T + 2 et T - 2 dans une certaine mesure. Mais sur ces différents top on doit en choisir un et un seul. La présence de la 15ème variable permet donc de choisir automatiquement un seul de ces tops, par le programme d'apprentissage lui-même.

Lors de l'apprentissage de la structure, cette variable a son état déterminé par la durée écoulée depuis le dernier top du professeur. Lors des tests de la structure, son état représente par contre, la durée écoulée depuis le dernier top détecté par l'algorithme lui-même.

Nous avons donc étendu à une petite échelle l'apprentissage automatique de structures à de la programmation dynamique.

Remarque 2

A la structure obtenue par l'apprentissage correspond une hiérarchie sur les variables d'entrée qui regroupe en priorité les variables proches dans le plan temps fréquence :

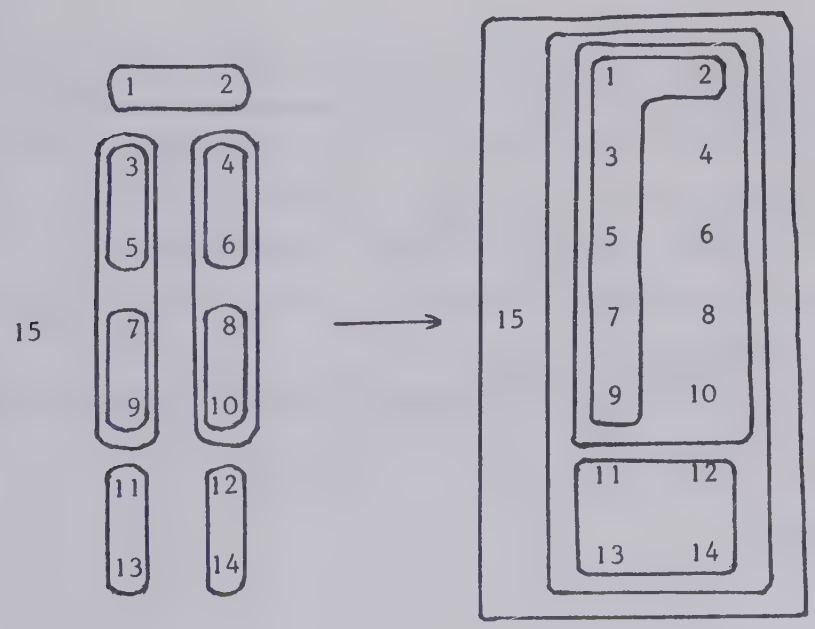


Figure 39

L'opérateur au sommet de la hérarchie a comme variables d'entrée, A_{15} et A_{28}. Cette dernière est la variable de sortie d'une structure construite sur les 14 premières variables. A_{15} mesure la durée écoulée depuis le dernier top. La table de cet opérateur est la suivante :

A_{28} \ A_{15}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
2	0	0	0	0	0	1	1	0	1	1	1	1	1	0	1
3	0	0	0	1	0	0	0	1	1	1	0	1	0	1	1
4	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1
5	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 40

Cette table montre bien l'influence du contexte sur le résultat.

Malheureusement, les autres tables d'opérateurs ont moins de signification. En effet, les tables créées ne sont ni la table d'addition ni la table de multiplication, et au vu de ces seules tables on ne peut rien déduire quant à la manière dont le phénomène a été analysé.

La structure est créée, on connait ses performances, mais on ne sait pas comment la reconnaissance s'effectue.

C'est un inconvénient de la méthode d'apprentissage. En particulier les acousticiens qui s'occupent de reconnaissance de la parole s'intéressent plutôt à la manière dont on peut reconnaître la parole, qu'à la reconnaissance elle-même.

Remarque 3

Sur ce même problème, les méthodes d'apprentissage utilisant la séparation par hyperplans ont donné des résultats peu utilisables (taux de réussite de 17 % sur l'ensemble d'apprentissage).

Remarque 4

Nous avons étudié l'influence des importances relatives des différentes formes sur les performances de la structure finale. Dans les expériences suivantes, il n'y a pas de programmation dynamique : les variables d'entrée sont au nombre de 14. Les top étant détectés par séries le critère de choix est de prendre le premier de chaque série. De plus le paramètre n_r de l'algorithme d'apprentissage fixant le nombre d'états des variables intermédiaires, a comme valeur 8 alors que la valeur définitive de n_r vaut 6 (performances déjà présentées).

Importances relatives	Top détectés	Top non détectés	Top en trop
1 et 1	3	51	1
1 et 8	26	28	11
1 et 10	43	14	17

Figure 41

(Ces performances sont mesurées sur l'ensemble de test).

VI.6. RECONNAISSANCE DE MOTS A PARTIR DE PHONEMES : PRETRAITEMENT (DONNES FOURNIES PAR UN SYSTEME DE RECONNAISSANCE DE PHONEMES)

VI.6.1. Introduction du problème

Nous avons étudié le problème de la reconnaissance de mots parlés en supposant le problème de la reconnaissance de phonèmes résolu avec une certaine probabilité d'erreur. Nous l'avons étudié avec des idées voisines de celles qui ont servi de base au programme décrit précédemment sans tout-à-fait utiliser ce programme.

Les mots à reconnaître sont au nombre de 450. Cette liste de mots a été établie par le Centre National d'Etudes des Télécommunications : voir [23].

Les phonèmes supposés reconnus sont les suivants :

Figure 42

Code	Phonème	Code	Phonème
1	A	17	L
2	E bref (cheval)	18	M
3	I	19	N
4	O (auto)	20	P
5	U	21	R
6	E (donnée)	22	S (silence)
7	AN (antérieur)	23	T
8	IM (important)	24	V
9	OU	25	CH (cheval)
10	OI (choix)	26	G (gué)
11	JJ (mouillé)	27	Z (oisif)
12	B	28	AI (lettre)
13	D	29	ON (chanson)
14	F	30	EU (heuristique)
15	J (journal)	31	W (point)
16	K (caractère)	32	AU (horloge)

L'étage de reconnaissance de phonèmes n'étant pas encore au point et surtout n'étant pas apte à fournir des données suffisantes pour remplir correctement les tableaux de probabilités, il a été remplacé par un programme de simulation. Celui-ci prend chacun des 450 mots au départ phonétiquement correct et y introduit des erreurs du même type que celles du programme de reconnaissance de phonèmes : des phonèmes sont supprimés, d'autres rajoutés, d'autres enfin mal reconnus. Le taux d'erreur de l'étage de reconnaissance a été établi à une valeur supérieure à 20 % de telle manière que l'homme peut encore reconnaître les mots prononcés avec une probabilité de l'ordre de 0,98.

Le but du problème est donc de trouver un programme décidant que les deux prononciations (par exemple) :

P L O É S E

C L O F É S E

sont deux prononciations du mot P R O F E S S E U R.

On remarque ici qu'il existe certains opérateurs apparamment informants, tels :

O_5 = [Quel est le phonème prononcé le 5ème ?]

Malheureusement, dans le précédent exemple, les réponses de O_5 sont respectivement S et E : il est plus judicieux de poser la question :

[Quel est le phonème situé après le premier E du mot ?]

Dans les deux cas, la réponse est S. Il faut donc rechercher l'invariance des opérateurs par rapport à leur position dans le mot.

VI.6.2. Les opérateurs diphonèmes simples

Les opérateurs choisis dans un premier temps sont les 32 x 32 = 1 024 opérateurs :

$O_{\varphi_1 < \varphi_2}$ = {existe-t-il dans le mot les phonèmes φ_1 et φ_2 tels que φ_1 précède φ_2 dans ce mot ?}

Ce sont des opérateurs à deux réponses.

Nous avons mesuré l'information de ces opérateurs et nous les avons classés par ordre d'information décroissante. Voir figure 43 et 44. (ces figures représentent le même graphique avec deux échelles différentes).

Le meilleur opérateur est celui qui est associé aux phonèmes R et E. Son information utile est de 0,39, à comparer avec l'information de l'opérateur idéal : $H[\Omega] = 8,7$.

Remarque :

Le calcul de la fonction information utile est fait sur un univers d'apprentissage comprenant 100 échantillons de chaque mot soit un total de 45 000 mots introduits au moyen du programme de simulation de l'étage de reconnaissance de phonèmes.

VI.6.3. Les opérateurs diphonèmes améliorés

Nous allons considérer à présent les positions respectives des différents phonèmes. Un mot à reconnaître est formé de phonèmes φ dont les positions sont repérées par la variable X. Il y a une probabilité négligeable pour qu'un mot à reconnaître ait plus de 12 phonèmes (dans le problème considéré). X a donc 12 états différents, φ en a 32.

1 INFO(R E)= .39648E+00
2 INFO(T E)= .33646E+00
3 INFO(A R)= .31864E+00
4 INFO(I R)= .30619E+00
5 INFO(R I)= .29599E+00
6 INFO(T I)= .28576E+00
7 INFO(A T)= .28186E+00
8 INFO(A E)= .28150E+00
9 INFO(I E)= .27099E+00
10 INFO(A I)= .27009E+00
11 INFO(K E)= .26249E+00
12 INFO(A L)= .25620E+00
13 INFO(I S)= .25386E+00
14 INFO(L E)= .24948E+00
15 INFO(I T)= .24431E+00
16 INFO(D I)= .23417E+00
17 INFO(AI E)= .23179E+00
18 INFO(K T)= .23028E+00
19 INFO(AI R)= .23022E+00
20 INFO(S T)= .23012E+00
21 INFO(S E)= .22578E+00
22 INFO(S I)= .22443E+00
23 INFO(AN T)= .22102E+00
24 INFO(A S)= .22161E+00
25 INFO(D E)= .21949E+00
26 INFO(M E)= .21635E+00
27 INFO(K R)= .21623E+00
28 INFO(R T)= .21515E+00
29 INFO(K A)= .21454E+00
30 INFO(P R)= .21201E+00
31 INFO(M I)= .21156E+00
32 INFO(K S)= .19835E+00
33 INFO(IJJ)= .19820E+00
34 INFO(AN E)= .19513E+00
35 INFO(R S)= .18971E+00
36 INFO(K U)= .18854E+00
37 INFO(I R)= .18727E+00
38 INFO(R A)= .18558E+00
39 INFO(J E)= .18399E+00
40 INFO(I L)= .18306E+00
41 INFO(R R)= .18267E+00
42 INFO(SAN)= .17782E+00
43 INFO(AEE)= .17732E+00
44 INFO(REE)= .17697E+00
45 INFO(A A)= .17652E+00
46 INFO(A K)= .17651E+00
47 INFO(I I)= .17627E+00
48 INFO(O E)= .17611E+00
49 INFO(I N)= .17454E+00
50 INFO(P E)= .17344E+00
51 INFO(O R)= .17334E+00
52 INFO(U E)= .17332E+00
53 INFO(Z E)= .17331E+00
54 INFO(K I)= .17322E+00
55 INFO(EE R)= .17295E+00
56 INFO(TEE)= .17273E+00
57 INFO(EE I)= .17259E+00

Figure 43-a

```
 63  INFO( RAN)= .164641+00  *
 64  INFO( S S)= .163361+00  *
 65  INFO( M T)= .161265+00  *
 66  INFO( SJJ)= .160708+00  *
 67  INFO( L T)= .160437+00  *
 68  INFO(EE A)= .159843+00  *
 69  INFO( U T)= .156929+00  *
 70  INFO( LEE)= .156064+00  *
 71  INFO( I K)= .155249+00  *
 72  INFO( TAN)= .154354+00  *
 73  INFO( R M)= .154134+00  *
 74  INFO( R L)= .153705+00  *
 75  INFO( M A)= .153384+00  *
 76  INFO(JJUN)= .152998+00  *
 77  INFO( T U)= .152749+00  *
 78  INFO( U I)= .152516+00  *
 79  INFO( RAI)= .152288+00  *
 80  INFO( O I)= .151316+00  *
 81  INFO( K L)= .149584+00  *
 82  INFO( K O)= .149530+00  *
 83  INFO( P I)= .148095+00  *
 84  INFO( KON)= .147554+00  *
 85  INFO( E R)= .147466+00  *
 86  INFO( G R)= .146945+00  *
 87  INFO( N E)= .146893+00  *
 88  INFO( P S)= .146296+00  *
 89  INFO( L R)= .146190+00  *
 90  INFO( T T)= .145922+00  *
 91  INFO( P A)= .143878+00  *
 92  INFO(EE S)= .143824+00  *
 93  INFO( O S)= .142717+00  *
 94  INFO( M L)= .142043+00  *
 95  INFO( PEE)= .140384+00  *
 96  INFO( KJJ)= .140229+00  *
 97  INFO(EE L)= .139773+00  *
 98  INFO( G E)= .138491+00  *
 99  INFO( V E)= .138328+00  *
100  INFO( T K)= .137759+00  *
101  INFO( I A)= .137416+00  *
102  INFO( O O)= .137367+00  *
103  INFO( S K)= .137332+00  *
104  INFO( MAN)= .135247+00  *
105  INFO( U L)= .135216+00  *
106  INFO( L T)= .135170+00  *
107  INFO( TAI)= .134302+00  *
108  INFO( B E)= .134300+00  *
109  INFO( SON)= .134217+00  *
110  INFO( T S)= .132485+00  *
111  INFO(EE E)= .131428+00  *
112  INFO( I U)= .131292+00  *
113  INFO( MEE)= .131214+00  *
114  INFO( T A)= .130488+00  *
115  INFO( U R)= .130346+00  *
116  INFO( U S)= .130094+00  *
117  INFO( R Z)= .129632+00  *
118  INFO( D A)= .128893+00  *
119  INFO(IM E)= .128360+00  *
120  INFO( F E)= .128179+00  *
```

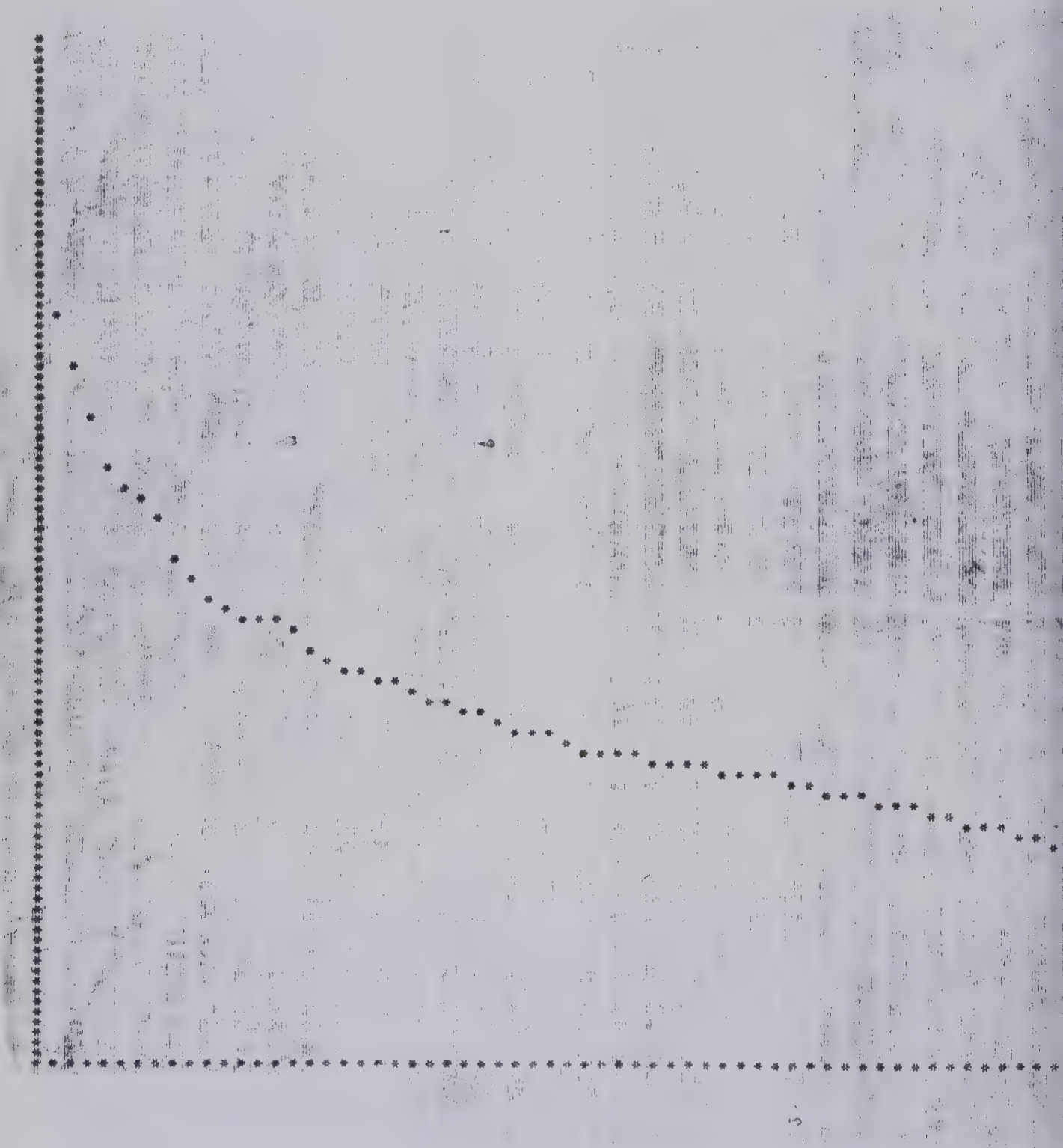


Figure 44-a

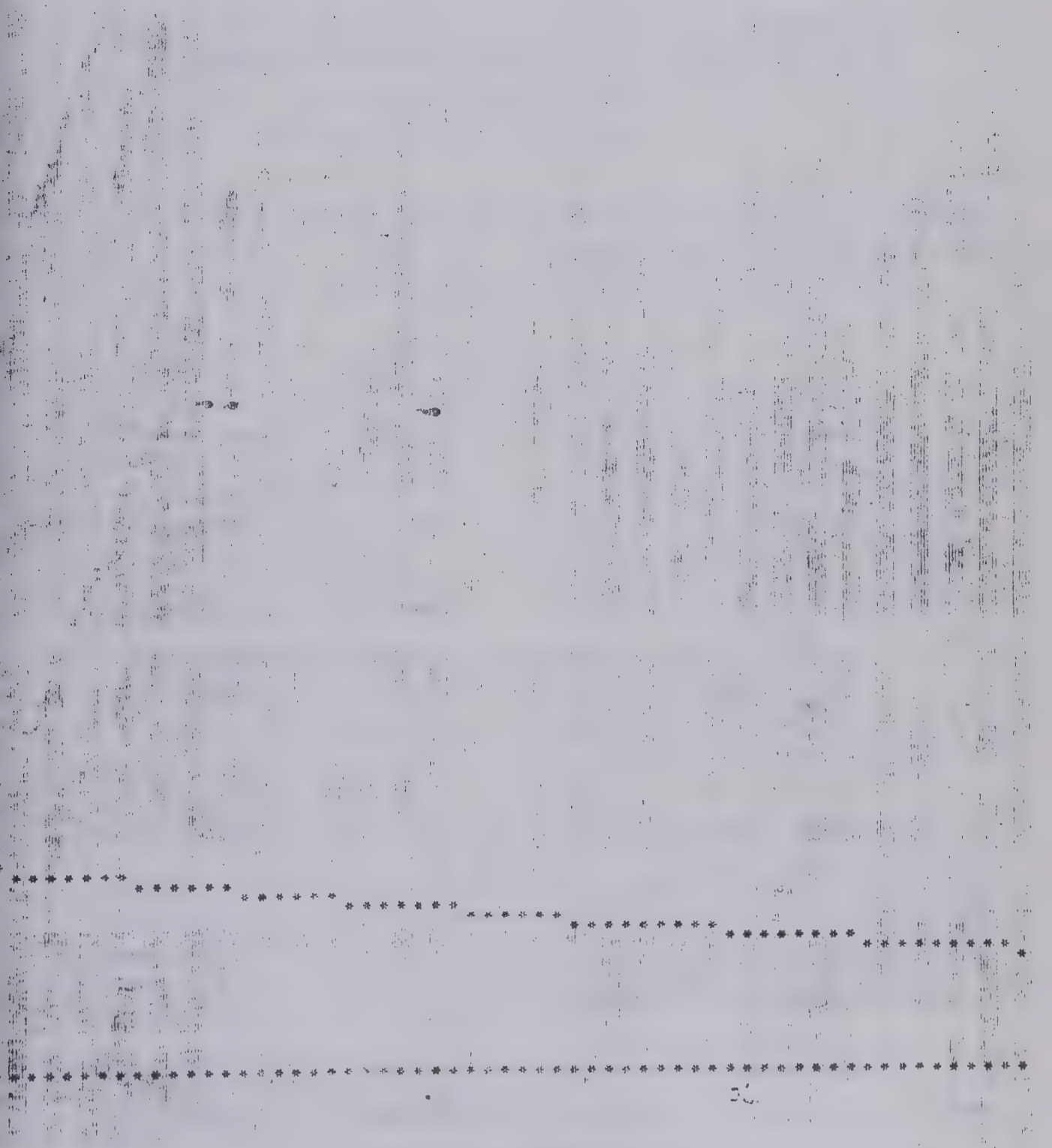


Figure 44-b

A un couple de phonèmes ordonnés (φ_1, φ_2) correspond $\frac{12 \times 11}{2} = 66$ valeurs possibles du couple (X_1, X_2) de leurs positions. Au couple (φ_1, φ_2) on peut donc faire correspondre les 66 opérateurs différents :

$$\{O_{\varphi_1, \varphi_2, i} = [\text{existe-t-il dans le mot le couple de phonèmes } (\varphi_1, \varphi_2) \text{ à la position i ?}] \mid i \in (1, \ldots, 66)\}$$

chacun de ces opérateurs a comme variable de sortie $A_{\varphi_1, \varphi_2, i}$, dont les états sont {oui, non} ou {0, 1}, ...

Notons le produit cartésien de ces 66 ensembles :

$$A_{\varphi_1, \varphi_2} = \prod_{i=1}^{66} A_{\varphi_1, \varphi_2, i}$$

Nous considérons cette nouvelle variable comme la variable de sortie de l'opérateur appelé :

$$O_{\varphi_1, \varphi_2} = \prod_{i=1}^{66} O_{\varphi_1, \varphi_2, i}$$

$$= \prod_{i=1}^{66} [\text{existe-t-il dans le mot le couple de phonèmes } (\varphi_1, \varphi_2) \text{ en i ?}]$$

le nombre d'états des variables A_{φ_1, φ_2} est grand (2^{66}) : il est trop grand pour que ces variables puissent servir d'entrée à un programme d'apprentissage du type décrit précédemment. Aussi avons nous créé, de nouveaux opérateurs. Ceux-ci sont dérivés des O_{φ_1, φ_2} par la même méthode de regroupement que celle du programme d'apprentissage.

Le regroupement s'effectue, non en créant une partition sur l'ensemble A_{φ_1, φ_2}, maix sur l'ensemble des 66 positions (X_1, X_2) des phonèmes φ_1 et φ_2. Il lui correspond un opérateur $Y \leftarrow Pos(X_1, X_2)$. Si ν est le nom-

bre d'états de Y, il existe ν opérateurs tels que :

$$O'_{\varphi_1, \varphi_2, Y} = [\text{existe-t-il } \varphi_1 \text{ et } \varphi_2, \text{ en } X_1 \text{ et } X_2 \text{ tels que } Pos(X_1, X_2) = Y \ ?]$$

D'où de la même façon que précédemment.

$$O'_{\varphi_1, \varphi_2} = \prod_{Y=1}^{\nu} O'_{\varphi_1, \varphi_2, Y}$$

dont la variable de sortie est $A'_{\varphi_1, \varphi_2}$.

Si nous prenons $\nu = 4$, le nombre de réponses de l'opérateur $O'_{\varphi_1, \varphi_2}$ est égal à $2^4 = 16$.

Le regroupement sur l'ensemble des positions (X_1, X_2) s'effectue avec la méthode décrite en VI.2. en optimisant l'information utile transmise par l'opérateur $O'_{\varphi_1, \varphi_2}$. A chaque itération de la méthode, le nombre ν décroit d'une unité, le nombre de réponses de $O'_{\varphi_1, \varphi_2}$ est divisé par 2.

La figure 46 montre le regroupement obtenu sur (X_1, X_2) en utilisant les phonèmes R et E, c'est-à-dire la table de l'opérateur $Y \leftarrow Pos(X_1, X_2)$ correspondant aux positions de R et E. La figure 45 montre l'évolution de l'information utile de $O'_{R, E}$ en fonction du numéro de l'itération de la méthode de regroupement.

L'opérateur $Y \leftarrow Pos(X_1, X_2)$ a quelque ressemblance avec $Y \leftarrow (X_1 - X_2)$ (opération soustraction) dans la zone où $X_1 - X_2 \leqslant 4$. Ce fait est plus ou moins intuitif, dans la mesure où la position relative des deux phonèmes est plus invariante que leurs positions absolues, après les erreurs dues à l'étage de reconnaissance de phonèmes (cf. l'exemple des deux représentations du mot P R O F E S S E U R).

**********COUPLE DE PHONEMES CONSIDERE :   R   E   ***********
****

CE GRAPHIQUE REPRESENTE LA VARIATION DE L'INFORMATION DURANT LE REGROUPEMENT

INFORMATION AVANT LE REGROUPEMENT : .9491

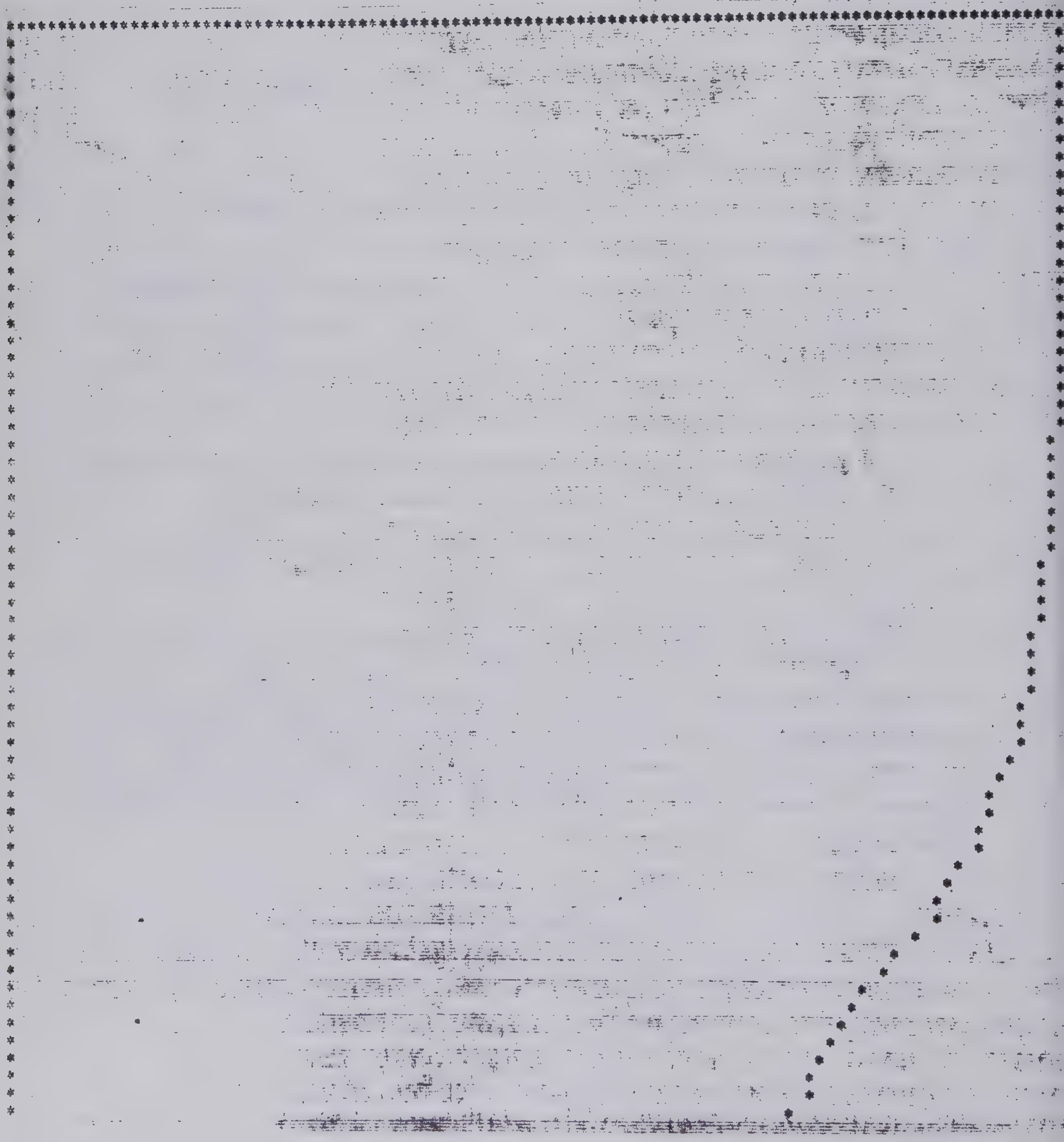


TABLEAU DES RECOUPEMENTS
INFORMATION APRES RECOUPEMENT (RECALCULEE) : .7022
(OBTENUE PAR LA SOMME DES PERTES D'INFORMATION : .7022)

figure 46

On obtient expérimentalement pour les phonèmes R et E, les informations utiles :

$$I[A_{\varphi_1 < \varphi_2} ; \Omega] = 0,39$$

$$I[A_{\varphi_1, \varphi_2} ; \Omega] = 0,94$$

$$I[A'_{\varphi_1, \varphi_2} ; \Omega] = 0,70$$

(à comparer à l'information utile idéale de 8,7 pour la reconnaissance des 50 mots).

On constate donc une diminution de l'information utile par rapport à $I[A_{\varphi_1, \varphi_2} ; \Omega]$, mais peu importante si on considère la diminution du nombre de réponses de l'opérateur de 2^{66} à 16.

Ce nombre de réponses permet alors l'utilisation des variables $A'_{\varphi_1, \varphi_2}$ comme variables d'entrée d'une structure de reconnaissance engendrée par notre programme général d'apprentissage.

VI.6.4. Factorisation des opérateurs de position

Il existe 1 024 couples de phonèmes correspondant à 1 024 opérateurs $O'_{\varphi_1, \varphi_2}$. Au lieu de garder en mémoire les 1 024 opérateurs $Y \leftarrow Pos(X_1, X_2)$, nous avons créé 50 de ces opérateurs en choisissant aléatoirement les 50 couples de phonèmes correspondants. Un programme détermine ensuite pour chacun des 1 024 couples de phonèmes, celui des 50 opérateurs de position qui donne l'information maximum pour l'opérateur $O'_{\varphi_1, \varphi_2}$ correspondant.

La figure 47 montre le classement des opérateurs $O'_{\varphi_1, \varphi_2}$ obtenu. On y indique le couple de phonème et le numéro de l'opérateur de position qui a été le meilleur pour ce couple.

Figure 47-a

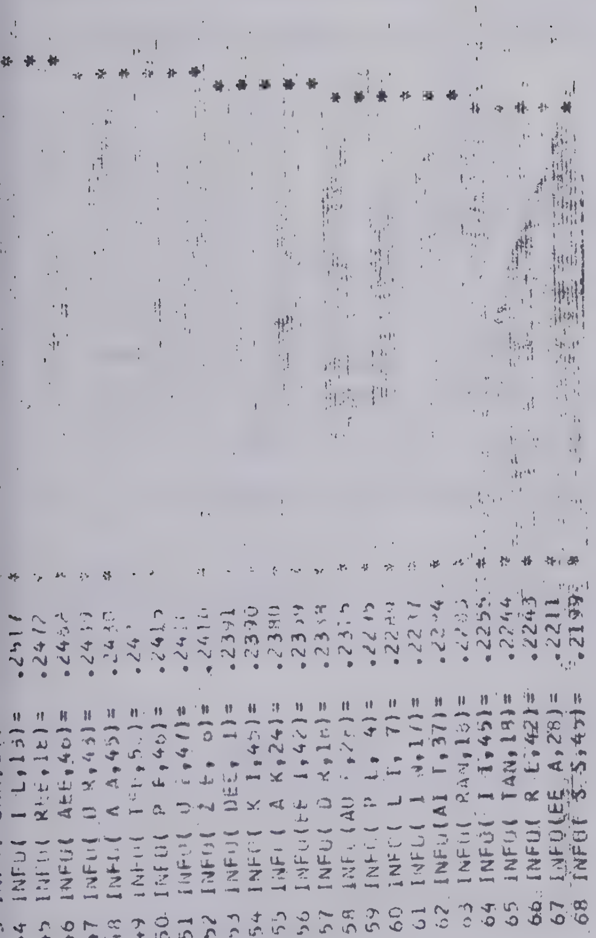

44 INFO(I L,13)= .2517
45 INFO(REE,1E)= .2472
46 INFO(AEE,46)=
47 INFO(O R,43)=
48 INFO(A A,45)=
50 INFO(P F,46)=
52 INFO(Z E, 6)=
53 INFO(DEE, 1)= .2391
54 INFO(K I,45)= .2390
55 INFO(A K,24)= .2380
56 INFO(EE I,42)=
57 INFO(D R,16)=
59 INFO(P L, 4)=
60 INFO(L I, 7)=
61 INFO(I N,17)=
62 INFO(AI T,37)=
63 INFO(RAN,18)=
64 INFO(I I,45)= .2255
65 INFO(TAN,18)= .2244
66 INFO(R E,42)= .2243
67 INFO(EE A,28)= .2211
68 INFO(S S,45)= .2199

Figure 47-b

L'occupation mémoire de l'ensemble des 50 opérateurs de position est donc faible (de l'ordre de 6 400 bits) et l'information utile des $O'_{\varphi_1, \varphi_2}$ n'a guère baissé.

VI.6.5. L'étage de reconnaissance de mots

Nous disposons maintenant de 1 024 opérateurs $O'_{\varphi_1, \varphi_2}$ dont les variables peuvent servir d'entrée à une structure engendrée par un programme d'apprentissage du type du § VI.3. L'apprentissage est rendu difficile par le nombre des variables d'entrée.

Les résultats semblent encourageants. L'information utile apportée par l'ensemble des quatre opérateurs les plus informants (O'_{RE}, O'_{TE}, O'_{AR}, O'_{TR}) est égale à 1,9 : elle est peu inférieure à la somme des informations utiles des opérateurs pris séparément.

Ces méthodes d'apprentissage peuvent sembler lourdes, mais les opérateurs engendrés réaliseront une reconnaissance extrèmement rapide et occuperont relativement peu de mémoire : 3 200 mots de 36 bits environ. Ces opérateurs sont en même temps beaucoup plus efficaces qu'une simple méthode de comparaison à un dictionnaire.

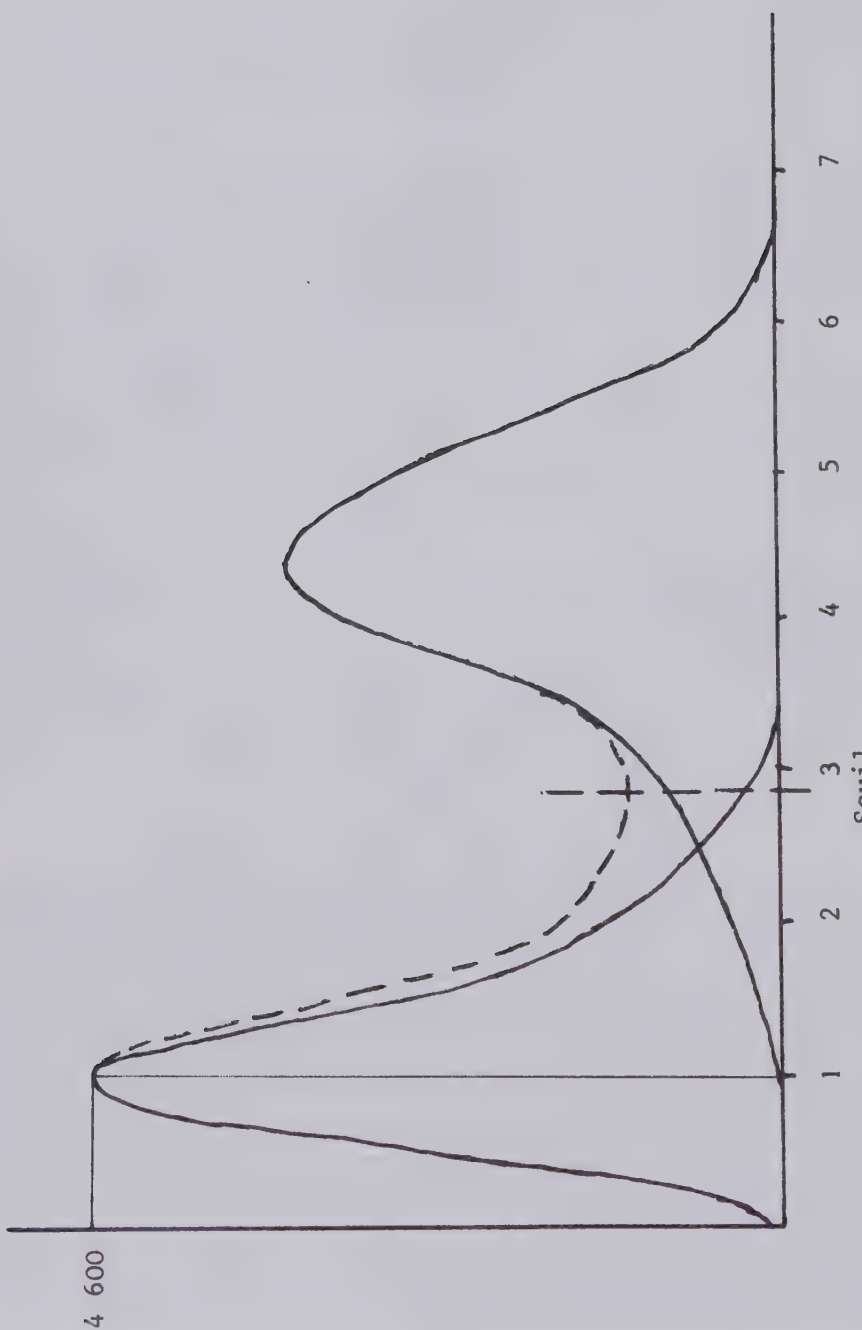
VI.7. RECONNAISSANCE DES FORETS SUR PHOTOS AERIENNES

Des photos aériennes ont été digitalisées par le capteur optique CALIFE du Laboratoire Central de l'Armement. Ce capteur délivre de chacune de ces photos une matrice de 480 sur 720 échantillons de luminosité. Chaque échantillon est codé sur 8 bits d'où 256 niveaux de luminosité différents possibles.

Les photos représentent des régions de champs et de forêts. Le but de notre problème est de déterminer par apprentissage un opérateur qui sépare les zones de forêts, des autres zones.

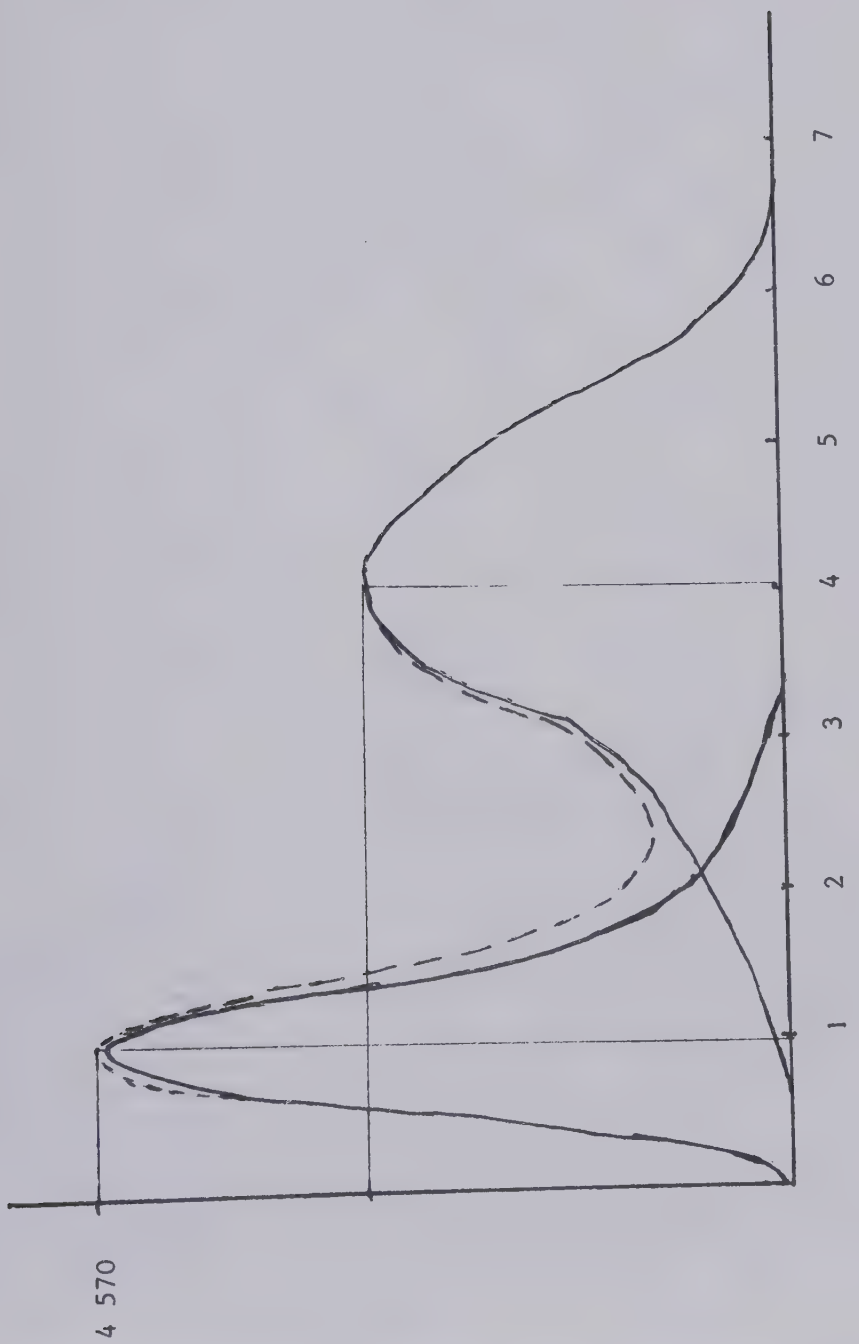
Nous avons travaillé dans un premier temps sur une seule de ces photos. La figure 48 montre les zones de forêts (astérisques) et les autres (blancs).

Figure 48 : Image initiale (reconstitution avec seuil)



Histogramme de la photographie

Figure 49



Histogramme de l'ensemble d'apprentissage

Figure 50

Figure 51 : Reconnaissance de la forêt après apprentissage sur les données non standardisées.

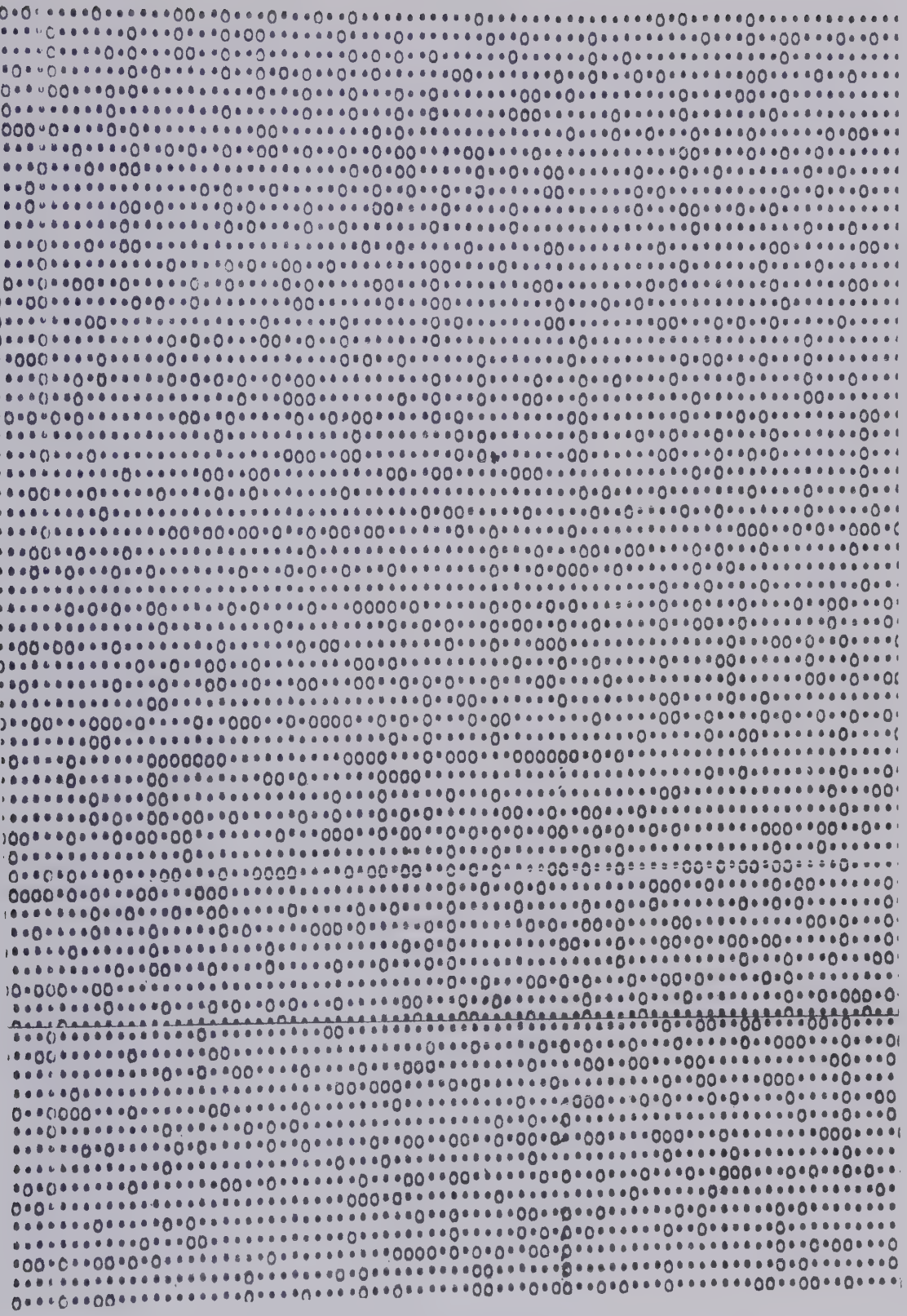


Figure 52 : photo standardisée (reconstitution avec seuil = 1,5)

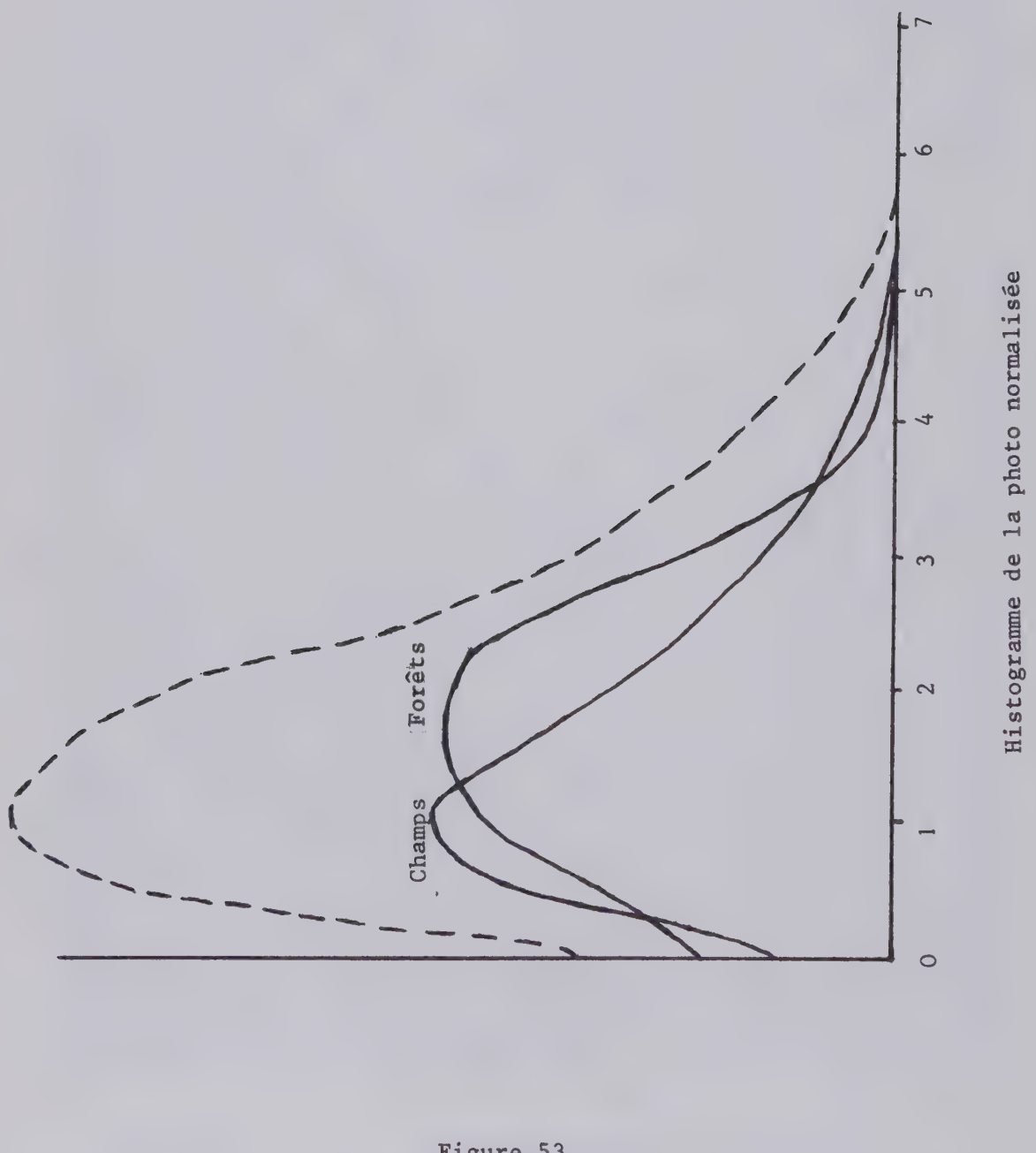

Forêts
Champs
0
1
2
3
4
5
6
7
Histogramme de la photo normalisée

Figure 53

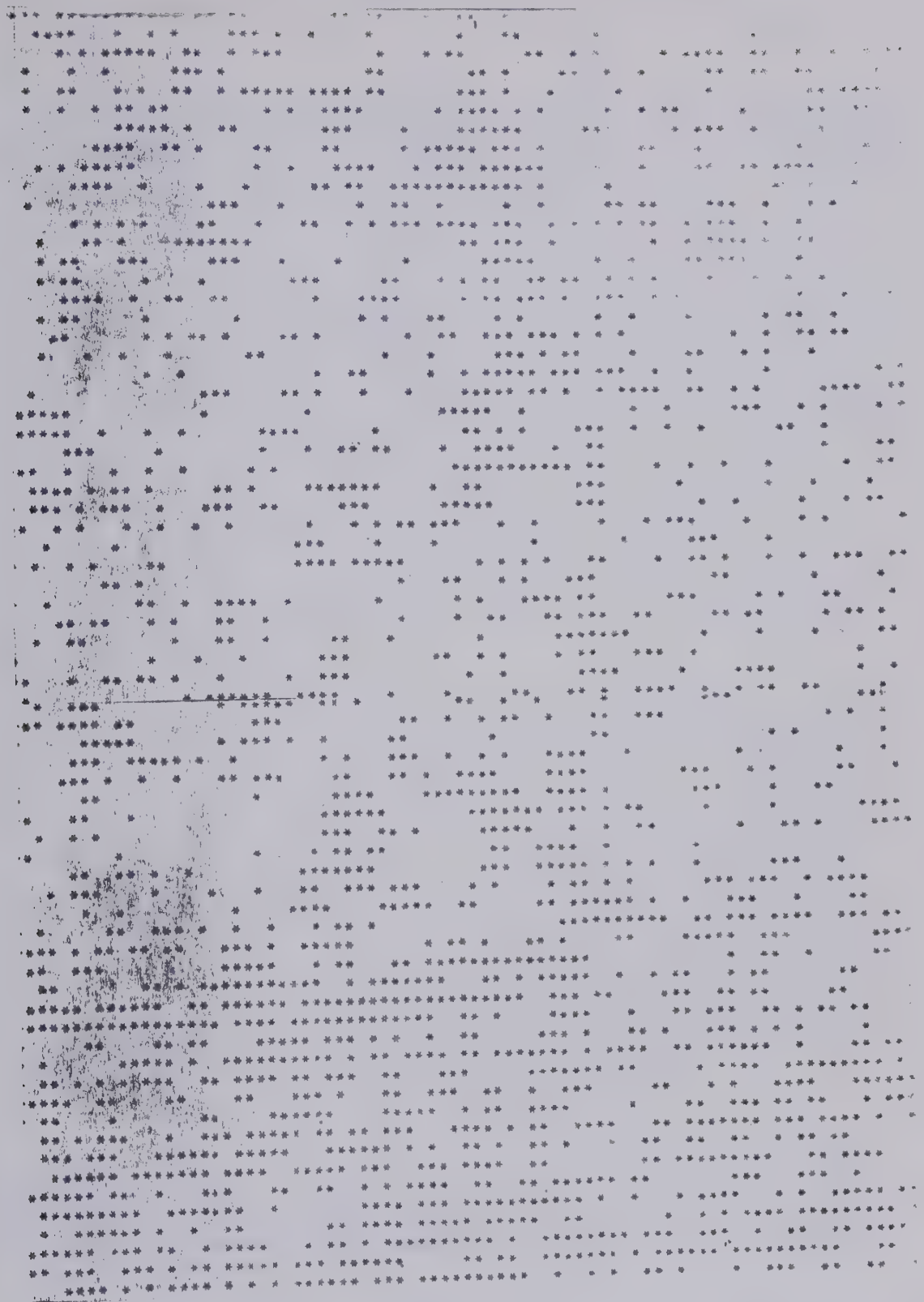


Figure 54 : Reconnaissance de la forêt à partir de la photo standardisée

Figure 55 : Lissage des résultats de la figure 54

Les données d'entrée que nous avons considérées sont les histogrammes locaux des luminosités aux différents points de la photo, en vue d'appliquer notre programme de génération de structure de reconnaissance.

La photo étudiée comporte 48 niveaux différents de luminosité (les 48 premiers niveaux des 256 que l'appareil de mesure CALIFE peut délivrer). L'intervalle [1, 48] est divisé en huit parties égales, ramenant le nombre de niveaux de la photo à huit. Les histogrammes comportent donc 8 variables A_j correspondant chacune à l'un de ces 8 niveaux. L'état de chaque variable est déterminé par le nombre d'éléments du niveau correspondans de luminosité dans la zone dont on veut calculer l'histogramme.

La figure 49 montre l'histogramme de la photo étudiée.

Les histogrammes locaux autour d'un point (x, y) sont calculés sur un carré centré sur (x, y) et de côté 3 éléments de maillage. D'où 9 éléments et 10 valeurs différentes possibles pour chacune des 8 variables d'entrée A_j.

Sur les 480 x 720 $\simeq$ 350 . 600 points de la photo, 10 000 ont été pris aléatoirement en vue de constituer l'ensemble d'apprentissage. Les autres points formeront l'ensemble de test. La figure 50 montre la moyenne des histogrammes locaux autour des points d'apprentissage.

1. Notre programme a été appliqué tel quel à l'ensemble des histogrammes locaux autour des points (x, y). Il a engendré la structure de reconnaissance travaillant sur les huit variables de l'histogramme. Nous l'avons testé sur la photo entière : figure 51 . L'information utile de la structure obtenue est de 0,678 (à comparer à $H[\Omega] = 0,684$). Le pourcentage d'erreur sur la photo totale est de 2 %.

2. Les résultats obtenus sont apparemment satisfaisants. Malheureusement, il faut tenir compte expérimentalement des dérives de l'appareil, des variations de gamma, d'une photographie à l'autre et même d'un bord à l'autre d'une même photographie. Aussi est-il nécessaire de standardiser la photo en éliminant la composante continue de la luminosité, et en uniformisant le contraste de la photo : on a divisé la matrice représen-

tive de la photo en carrés de 10 x 10 éléments. Puis, de manière à avoir sur tous les carrés une valeur moyenne $\overline{X}$ et un écart type σ constants, on a déterminé sur chaque carré, le couple (σ, x) tel que :

$$\begin{cases} \frac{1}{100} \sum_{i=1}^{100} \alpha(x_i - x) = \overline{X} \\ \frac{1}{100} \sum_{i=1}^{100} (\alpha(x_i - x) - \overline{X})^2 = \sigma^2 \end{cases}$$

où les x_i sont les valeurs de la photo numérisée initiale et les $\alpha(x_i - x)$ les valeurs de la photo normalisée.

La figure 52 montre la photo ainsi normalisée. On remarque qu'on ne reconnaît plus à l'oeil les zones de forêt.

La figure 53 montre l'histogramme de la photo normalisée. On voit que les histogrammes de la forêt et du champ sont imbriqués de manière à interdire la décision par seuil.

Nous avons appliqué sur cette photo normalisée, le programme d'apprentissage. La structure engendrée a une information utile de 0,471 ($H[\Omega] = 0,684$). Le pourcentage d'erreur sur la photo est maintenant de 10 % : figure 54.

Après un opérateur heuristique de lissage (opérateur linéaire) on obtient la reconstitution de la figure 55.

Remarque 1

Au lieu de travailler sur les histogrammes nous avions dans un premier temps pris comme variables d'entrée de la structure de reconnaissance à engendrer, les 9 variables donnant la luminosité des points du carré 3 x 3 entourant le point (x, y). La reconnaissance sur la photo non standardisée avait les mêmes performances qu'en utilisant les variables définissant l'histogramme local au point (x, y).

Par contre, la photo standardisée amenait à des performances plus aibles. Il fallait itérer le processus d'apprentissage 3 fois et faire un issage final pour obtenir le même résultat qu'après un seul passage de 'algorithme d'apprentissage utilisant l'histogramme local.

Remarque 2

L'histogramme local est calculé sur un carré 3 x 3. Pour des performances supérieures il sera sans doute plus avantageux de travailler sur des carrés 4 x 4 ou 5 x 5. Les histogrammes seront alors plus précis et surtout plus représentatifs.

CONCLUSION

Dans ce problème, entièrement différent des problèmes de reconnaissance vocale, on a réussi à obtenir par apprentissage avec le même programme des résultats apparemment intéressants, puisque d'une photo apparemment illisible (la photo standardisée) le programme a extrait avec une bonne probabilité les formes à reconnaître.

VI.8. APPRENTISSAGE D'UNE NOTE DE RESSEMBLANCE

VI.8.1. Introduction

Nous présentons ici une autre utilisation du même programme de génération de structure, sur un problème qui n'est plus un problème de reconnaissance de formes au sens classique, visuel ou acoustique. C'est un problème d'apprentissage dans lequel il est presque impossible, par une simple réflexion de créer l'algorithme résolvant le problème : il faut ici approcher le plus possible un jugement de valeur subjectif par un opérateur automatique. Nous avons utilisé le même programme d'apprentissage, pour engendrer cet algorithme.

VI.8.2. Cadre du problème

Il s'agit d'une partie d'une étude sur le "recalage automatique sur image radar" faite par A. CHECROUN [9]. Les échos Radar des signaux émis par un avion déterminent sur le relief survolé des zones d'ombre. Connaissant les positions de celles-ci et de l'avion on peut simplement en déduire les positions des crêtes des montagnes donnant les zones d'ombres. Ce sont ces crêtes qui ont été prises comme formes primitives sur chaque image radar.

Un algorithme de filtrage digital [80] et [81] permet de donner de chacune de ces crêtes, une représentation par une succession de segments de longueurs inégales et juxtaposées que l'on détermine par les coordonnées de leur centre, leur position angulaire et leur longueur.

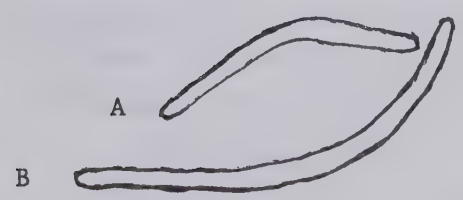


Fig. 56-a
Crêtes

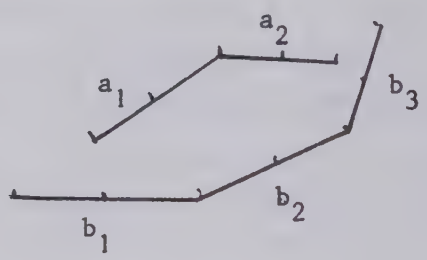


Fig. 56 -b
Crêtes après filtrage

A une crête peut correspondre différentes valeurs des paramètres suivant l'altitude de l'avion, sa direction... Il s'agit de trouver des classes de crêtes donnant lieu à une invariance par ces diverses transformations. Les travaux de E. DIDAY [15] ont conduit à un algorithme de classification performant que nous avons voulu utiliser. Cet algorithme nécessit

la définition d'une "distance de ressemblance" entre objets à classifier. Chaque objet est défini par un certain nombre de paramètres. DIDAY utilise la distance euclidienne, la distance du X^2, de Mahalanobis, etc... Mais il n'y a aucune raison que ces distances correspondent aux notions intuitives de ressemblance.

Nous avons voulu utiliser notre algorithme d'apprentissage pour créer un opérateur "distance de ressemblance" qui soit une bonne approximation des notions intuitives de ressemblance.

VI.8.3. Résolution du problème

Etant données deux crêtes, un observateur humain est capable d'ordonner une "note de ressemblance" c'est-à-dire un nombre compris entre 0 et 10 traduisant une impression purement intuitive de la ressemblance de ces deux crêtes.

Six paramètres définiront chaque crête pour le calcul de la distance de ressemblance : si A et B désignent les extrémités d'une crête et M le milieu de la ligne $\overset{\frown}{AB}$, (X_A, X_B, X_M, Y_A, Y_B, Y_M) les coordonnées de ces trois points, ces 6 paramètres représentent :

$X_B - X_A$
$X_M - X_A$
$Y_B - Y_A$
$Y_M - Y_A$
Longueur $\overset{\frown}{AB}$
Moyenne des distances des points de $\overset{\frown}{AB}$
au centre de gravité de la crête.

La structure d'opérateurs engendrée aura 12 variables d'entrée, les 6 premières définissant l'une des deux crêtes à comparer, les 6 autres définissant l'autre crête. L'ensemble Ω des "formes" à reconnaître est ici l'ensemble des notes de ressemblance possibles $\{1, 2, \ldots, 9\}$.

a. Nous avons appliqué l'algorithme d'apprentissage à un ensemble de 100 données. L'algorithme a ainsi engendré la structure représentée sur la figure [57]. Les nombres décimaux affectés à chaque variable de la structure sont les informations utiles contenues dans ces variables.

Nous montrons aussi la matrice de confusion de l'algorithme, calculée sur l'ensemble d'apprentissage lui-même. Les résultats sont apparemment très satisfaisants. Malheureusement dans cet essai, les données dont nous disposions étaient en nombre trop faible pour pouvoir être partagé en données d'apprentissage et données de test.

b. Nous avons donc dans une deuxième expérience utilisé 600 données. Elles ont été partagées en 500 données d'apprentissage et 100 données de test.

Le graphe de la structure obtenue est rigoureusement le même que le graphe de celle de l'expérience précédente. Leurs opérateurs par contre sont difficilement comparables.

La page 197 montre les deux matrices de confusion calculées sur les deux ensembles, d'apprentissage et de test.

VI.8.4. Conclusions

a. Il est intéressant de constater, que sur le tableau exprimant les performances réelles de l'algorithme engendré (données de test), 94 % des notes données par l'algorithme différent de 2 points des notes intuitives

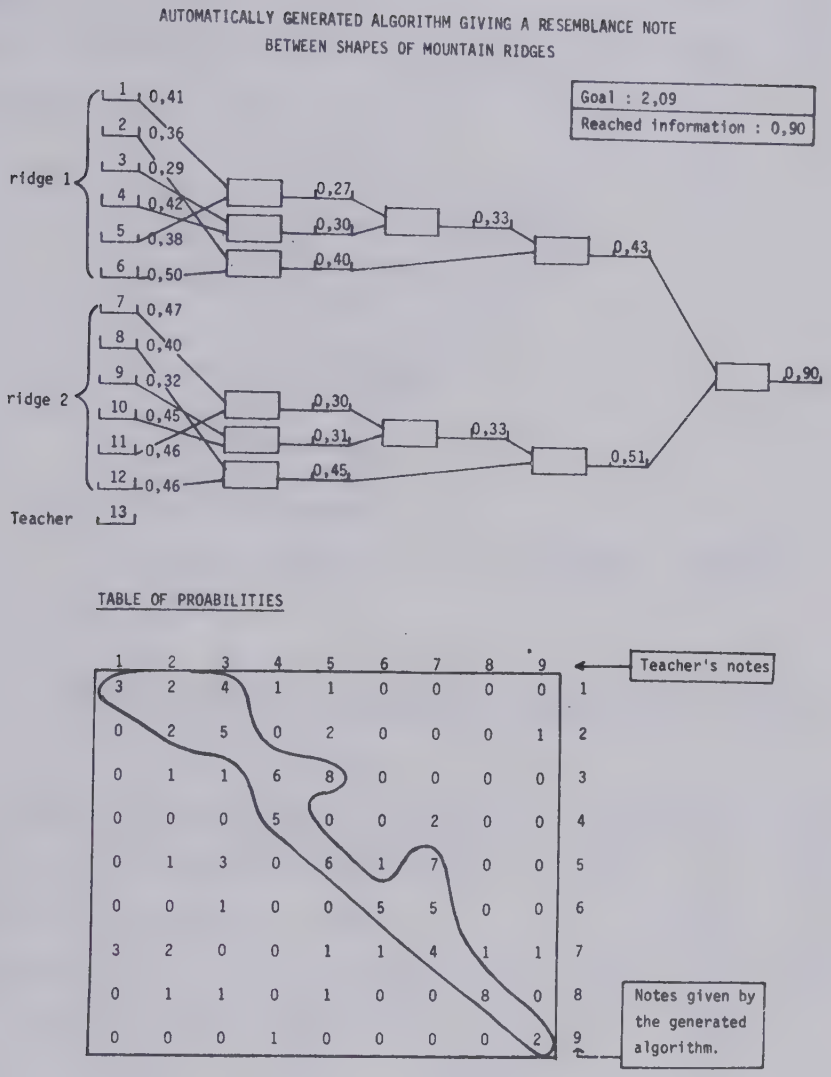


TABLE OF PROABILITIES

1	2	3	4	5	6	7	8	9	
3	2	4	1	1	0	0	0	0	1
0	2	5	0	2	0	0	0	1	2
0	1	1	6	8	0	0	0	0	3
0	0	0	5	0	0	2	0	0	4
0	1	3	0	6	1	7	0	0	5
0	0	1	0	0	5	5	0	0	6
3	2	0	0	1	1	4	1	1	7
0	1	1	0	1	0	0	8	0	8
0	0	0	1	0	0	0	0	2	9

Teacher's notes

Notes given by the generated algorithm.

Figure 57 : Expérience avec 100 données

Table des fréquences sur les 500 données d'apprentissage

	1	2	3	4	5	6	7	8	9
1	9	2	1	6	7	2	2	0	0
2	4	6	0	0	0	0	0	0	0
3	0	3	15	25	33	6	2	0	0
4	0	0	5	0	8	13	0	1	0
5	1	9	10	18	13	34	10	1	0
6	0	0	3	10	7	32	12	2	0
7	0	0	0	6	14	10	18	1	0
8	0	0	11	4	5	0	7	17	0
9	8	8	1	3	2	11	18	22	22

← Notes données par le "professeur"

↑ Notes données par la structure engendrée

Table des fréquences sur les 101 données de test

	1	2	3	4	5	6	7	8	9
1	0	0	0	1	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	1	2	8	10	1	0	0	0
4	0	0	1	0	3	7	0	0	0
5	0	2	3	3	4	11	2	0	0
6	0	0	0	6	4	14	7	0	0
7	0	0	0	0	2	2	3	0	0
8	0	0	0	1	1	0	0	1	0
9	0	0	0	0	0	0	0	0	0

← Notes données par le "professeur"

↑ Notes données par la structure engendrée

Figure 58

Expérience avec 600 données

données par le professeur : ces performances seraient semblables à celles d'un autre expérimentateur humain.

A partir des tableaux des informations utiles apportées par les différents facteurs, on déduit que le professeur en a tenu compte par l'ordre suivant d'importance :

$Y_M - Y_A$.
Longueur $\overset{\frown}{AB}$
$X_B - X_A$
Moyenne des distances...
$Y_B - Y_A$
$X_M - X_A$

Cet ordre est par ailleurs conforme à l'intuition.

b. L'étude du même problème sur deux ensembles d'apprentissage de dimensions différentes montre l'influence du nombre de données : les performances de la structure engendrée en premier sur les 100 données sont apparemment excellentes. Mais elles sont mesurées sur les données mêmes qui ont permis l'apprentissage, ce qui n'a aucun sens.

Les performances de la deuxième structure ont beaucoup plus de sens : d'une part, on considère des données de test différentes des données d'apprentissage, d'autre part, les performances mesurées sur les données de test et sur les données d'apprentissage sont semblables. Ce dernier point montre l'ergodicité des phénomènes étudiés, et la validité du nombre choisi d'expériences d'apprentissage.

c. Le graphe de la structure engendrée dans ce problème est symétrique par rapport aux deux crêtes. Ce fait est remarquable : cette symétrie qui se trouvait sans doute dans les données d'apprentissage elles-mêmes, a été extraite automatiquement par l'algorithme d'apprentissage.

Cette symétrie n'est pas due au fait que si un couple C_1 C_2 de crêtes est analysé dans les données, le couple C_2 C_1 l'est aussi. Nous avons en effet interdit les permutations de crête à l'intérieur des données d'apprentissage.

On peut expliquer partiellement cette symétrie par le fait que les couples C_1 C_2 sont en nombre suffisant pour que les informations mutuelles calculées sur les paramètres de C_1 et les mêmes calculées sur les paramètres de C_2 ont des valeurs voisines.

Il est aussi intéressant de constater que la structure engendrée conduit à une fonction "distance de ressemblance" de la forme :

$$D(C_1, C_2) = f(h_1(C_1), h_2(C_2))$$

h_1 et h_2 étant très voisines, et tendant vers la même fonction $h(C)$ lorsque le nombre de données d'apprentissage tend vers l'infini ; f étant une fonction voisine d'une fonction symétrique.

VI.9. Remarque sur la correspondance entre les formes et les réponses de la structure engendrée

L'opérateur engendré a pour réponses a(k), l'opérateur idéal $\omega(i)$. Le programme que nous avons présenté donne l'opérateur et la table de confusion entre les $\omega(i)$ et les a(k).

D'après nos hypothèses de travail, cette génération se fait indépendemment du nom des réponses a(k) et de la numérotation de celles-ci. Il faut donc établir une correspondance des a(k) dans les $\omega(i)$ telles que l'opérateur créé ait une probabilité d'erreur faible.

Si la structure de reconnaissance est parfaite :

$$I[A ; \Omega] = H[\Omega]$$

Nous avons vu que dans ce cas, il existe une correspondance biunivoque entre les $\omega(i)$ et les a(k). Il est alors facile au vu de la table de confusion d'établir cette correspondance et de faire les permutations nécessaires sur les a(k) pour obtenir une table diagonale.

Dans le cas où la table de confusion est non diagonalisable par permutations de ligne, nous avons utilisé un algorithme simple pour établir une correspondance satisfaisante :

Nous déterminons pour chaque ligne (chaque a(k)) la position de la plus grande valeur de la table de confusion soit i(k) cette position. C'est le numéro de la forme qui a la plus forte probabilité a posteriori après la réponse a(k). Nous ordonnons les a(k) suivant les i(k) croissant. Cet ordre donne la nouvelle numérotation des a(k).

Dans les exemples présentés, les tables de confusion sont toutes données transformées par cette nouvelle numérotation.

CONCLUSION ET PROSPECTIVE

Nous avons voulu associer dans cette étude des notions d'informatique (les opérateurs), de traitement de données et d'"intelligence artificielle" (heuristiques), dans le but de trouver des méthodes d'apprentissage en reconnaissance des formes.

Les résultats expérimentaux que nous avons obtenus avec nos deux méthodes se sont avérés très encourageants.

Nous continuons actuellement nos recherches dans cette direction en appliquant nos programmes à de nouveaux problèmes pratiques (reconnaissance de la parole, sonar, traitement d'images).

Au plan théorique, il serait sans doute intéressant d'associer la création d'opérateurs par leur table (2ème méthode) à la génération de la structure liant ces opérateurs par essai et sélection successifs ; utilisant ainsi à la fois les richesses des deux méthodes.

Une autre direction de recherche serait l'étude de la génération des opérateurs de commande en vue de s'adapter au mieux aux machines automatiques.

Enfin, l'utilisation plus poussée d'heuristiques permettrait d'accélérer la convergence des processus d'apprentissage dans les cas difficiles de génération de structures.

Ces études pourraient alors déborder du cadre strict de la reconnaissance des formes, pour s'appliquer aussi à l'apprentissage automatique de systèmes informatiques, d'automates, ou de programmes de simulation, sans passer par la création de modèles.

REFERENCES

[1] *G. J. BALM :* "An introduction to optical character reader considerations" Pattern Recognition. Vol. 2 - Nr 3 - Sept. 1970.

[2] *J. P. BENZECRI :* "Leçons sur l'analyse factorielle et la Reconnaissance des formes" Cours de l'ISUP (1965, 1966).

[3] *J. P. BENZECRI :* "Théorie de l'information et classification d'après un tableau de contingence" Publication interne de l'ISUP (1969).

[4] *D. BERTAUX :* "Méthodes mathématiques de la reconnaissance des formes" Rapport d'études du CERA, Octobre 1966.

[5] *M. BONGARD :* "Pattern Recognition" Spartan books 1970.

[6] *C. BROSSY :* "Les machines mathématiques" Science et Enseignement des sciences, n°28, Novembre - Décembre 1963.

[7] *S. CASTAN :* "Etudes expérimentales de discriminateurs linéaires" Colloque sur l'informatique, Toulouse, Mars 1966.

[8] *CHATEAUNEUF :* "Reconnaissance de photo aérienne par apprentissage" Rapport de diplôme d'Etudes Approfondies - Institut de Programmation - Juin 1972.

[9] *A. CHECROUN :* Thèse de Doctorat d'Etat - Paris - Mars 1972.

[10] *A. CHECROUN, C. ROCHE :* "Le Basic et le temps partagé"
Dunod 1970.

[11] *A. CHECROUN, C. ROCHE :* "La notion de squelette dans les problèmes de reconnaissance des formes"
Automatisme - Avril 1970.

[12] *M. F. DACEY :* "The syntax of a triangle and some other figures"
Pattern Recognition - Vol. 2 - Nr 1 - Janvier 1970.

[13] *M. DAVANCENS, M. HUET :* "Lecture automatique de lettres"
Diplôme d'Etudes Approfondies - Institut de Programmation - Juin 1970.

[14] *J. L. DELHAYE :* "Un programme de démonstration de théorèmes"
Thèse d'Université - Paris 1970.

[15] *E. DIDAY :* "Une nouvelle méthode en classification automatique et reconnaissance des formes : la méthode des nuées dynamiques"
IRIA - Document DIA : 445/ED - Février 1971.

[16] *E. DIDAY :* Thèse de Doctorat d'Etat
Paris 1972.

[17] *T. G. EVANS :* "A grammar controlled Pattern Analyser" - Information Processing 68 - North Holland - Publishing Company - Amsterdam (1969).

[18] *R. M. FANO :* "Transmission of Information"
MIT Press (1963).

[19] *A. FEINSTEIN :* "Foundations of Information Theory"
New-York, Mc Graw-Hill, 1958.

[20] *K. S. FU :* "Sequential Methods in Pattern Recognition and Machine Learning" Academic Press (1968).

[21] *V. GLUSHKOV :* "Automata Theory and Formal Microprogram Transformations" Kibernetika, vol. 1, n°5 (1965).

[22] *R. GODEMENT :* "Cours d'Algèbre" Hermann 1966.

[23] *J. Y. GRESSER :* Bulletin d'information du groupe communication parlée n°1 du GALF. Juin 1971, (supplément "Reconnaissance de la parole").

[24] *R. L. GRINSDALE, F. H. SUMMER, C. J. TUNIS, T. KILBURN :* "A system for the Automatic Recognition of patterns" Proc. Inst. of Electrical Engineers, Vol. 106, part. B, n° 26 - Mars 1959.

[25] *G. GUIHO :* Thèse de 3ème cycle - Paris 1971.

[26] *D. P. HATTAWAY, E. D. HIETANEN, R. W. ROTHFUSZ :* "Training a machine to read with non linear threshold logic" Electronics, Vol. 39, Nr 17, Août 1966. Mc Graw Hill.

[27] *J. C. HAY, F. C. MARTIN, C. W. WIGHTAM :* "The mark I perceptron design and performance" IRE International Convention record. Part. 2, Vol. 8, pp. 78-87, 1960.

[28] *J. C. HORVAT, R. HOUDAS :* "Représentation d'une fonction par une somme de translatées" Thèse de 3ème cycle - Juin 1970.

[29] *T. ITO :* "Mathematical Theory of programs and its applications" Journal of IEE in Japan (en Japonais) 1971.

[30] *T. ITO :* "On Glushkow's regular microprograms"
4th Hawaii International Conference on Systems Sciences (1971).

[31] *T. ITO :* "Super regular expressions and their applications to theory of programs"
Symposium on Mathematical Theory of Languages and Automata (1971) (en japonais).

[32] *J. P. JOUANNAUD :* "Filtres digitaux autoadaptatifs : algorithmes de calcul et simulation"
Thèse de doctorat de 3ème cycle - Février 1972.

[33] *E. KAHN :* Thèse de 3ème cycle - Paris 1971.

[34] *S. KAMAL ABDALI :* "Feature extraction algorithms"
Pattern Recognition - Vol. 3, Nr 1, Avril 1971.

[35] *L. KANAL, B. CHANDRASEKARAN :* "On dimensionality and sample size in statistical pattern classification"
Pattern Recognition - Vol. 3, Nr 3, October 1971.

[36] *A. I. KHINTCHINE :* "Mathematical foundations of Information Theory"
Dover 1957.

[37] *R. KIRSCH :* "Computer Interpretation of English Text and Picture Patterns"
IEEE - TEC, Vol. EC 13, Nr 4, Août 1964.

[38] *V. KOVALEWSKY :* "Character readers and Pattern Recognition"
Spartan Books 1968.

[39] *LAUZANNE :* "Reconnaissance automatique de mots"
Publication Interne - Institut de Programmation - Février 1972.

[40] *R. S. LEDLEY :* "FIDAC : Film input to digital automatic computer, and associated syntax directed pattern recognition programming system"
Symp. on Optical and Electrooptical Info. Proc. Tech. Boston, MIT Press 1965. Cf. T. GOLAB, R. S. LEDLEY, L. S. ROTOLO, Pattern Recognition, Vol. 3, Nr 2, Juillet 1971.

[41] *P. M. LEWIS :* "The characteristic Selection Problem in Recognition Systems"
IRE Trans. Infor. Theory, 8 pp. 171-178 (1962).

[42] *F. LIEGEOIS :* "La reconnaissance des formes"
Rapport d'études n° 2, DRME 63-34-145 - Centre d'Etudes et de Recherches en Automatisme, Paris Août 1964.

[43] *J. S. LIENARD :* "Analyse, Synthèse et Reconnaissance automatique de la Parole"
Thèse de Doctorat d'Etat - Avril 1972.

[44] *C. MACCHI :* "Itération stochastique et traitements numériques adaptatifs"
Thèse de Doctorat d'Etat - Juin 1972.

[45] *J. MAC QUEEN :* "Some Methods for classification and Analysis of Multivariate Observations"
5th Berkeley Symposium on Mathematics, statistics and probability, vol. 1, n°1, pp. 281-297.

[46] *C. G. MASSON :* "Hierarchical clustering of data with a mutual entropy criterion : its structural feature extraction ability"
IEEE Symposium on Syntactic Pattern Recognition, Chicago - Oct. 1970.

[47] *W. MEYER - EPPLER :* "Grundlagen und Anwendungen der Informations theorie"
Springer, Berlin, 1959.

[48] *O. N. MINOT :* "Automatic devices Recognitions of visible two dimensional Patterns. A survey of the field"
US Navy laboratory report 1050. San Diego, California (1961).

[49] *M. L. MINSKY :* "Steps towards Artificial Intelligence"
Proceedings IRE, vol. 49, n°1 - Janvier 1961.

[50] *M. L. MINSKY, S. PAPERT :* "Perceptrons"
MIT Press 1969.

[51] *K. I. MORI, H. GENCHI, S. WATANABE, S. KATSUROGI :* "Microprogram controlled pattern processing in a handwritten mail reader-sorter"
Pattern Recognition - Vol. 2, Nr 3 - Sept. 1970.

[52] *R. NARASIMHAN :* "Syntax directed interpretation of classes of pictures"
Comm. ACM Vol. 9, No 3, March 1966.

[53] *N. J. NILSSON :* "Learning machines"
Mc Graw-Hill - 1965.

[54] *M. PAVEL :* "Fondements mathématiques de la reconnaissance des structures"
Hermann - Paris 1969.

[55] *J. PFALZ, A. ROSENFELD :* "Web Grammars"
International Joint conference on Artificial Intelligence - Washington - Mai 1969.

[56] *C. F. PICARD :* "Theorie des Questionnaires"
Gauthier-Villars, 1965.

[57] *C. F. PICARD :* "Graphes et Questionnaires"
Gauthier-Villars, 1972.

[58] *E. RASEK :* "A contribution to the problem of feature selection with similarity functionals in pattern recognition"
Pattern recognition - Vol. 3, Nr. 1 - Avril 1971.

[59] *F. H. RAYMOND :* "Les principes des ordinateurs"
Presses Universitaires de France, 1969.

[60] *C. ROCHE :* "Optimisation d'un programme de reconnaissance de forme et détermination automatique de caractéristiques"
Publication interne - Institut de Programmation - Janvier 1971.

[61] *C. ROCHE :* "Evaluation des opérateurs de reconnaissance des formes et apprentissage"
Cours de l'Ecole Nationale Supérieure des Techniques Avancées.
Mai 1971.

[62] *C. ROCHE :* "Idées générales sur la reconnaissance des formes appliquées à la reconnaissance de la parole"
Congrès du Groupement des Acousticiens de Langue Française.
Aix-en-Provence - Mars 1971 - Publié dans Automatisme - Mars 1972.

[63] *C. ROCHE, G. SABAH :* "Définition d'une mesure de l'information utile d'un opérateur de reconnaissance. Génération automatique d'opérateurs de reconnaissance".
Compte rendu à l'Académie des Sciences - 7 Février 1972,
t. 274, p. 501-504.

[64] *C. ROCHE :* "Analyse linguistique selon WINOGRAD"
Journées d'Etudes sur la parole du Groupement des Acousticiens de Langue Française". Lannion - Juin 1970.

[65] *C. ROCHE :* "Génération automatique d'opérateurs de segmentation de la parole"
Journées d'Etudes sur la parole du Groupement des Acousticiens de Langue Française. Lannion - Juin 1972.

[66] *C. ROCHE, HEUDE :* "Lecture rapide et reconnaissance des formes"
Institut de Programmation - Juin 1970.

[67] *F. ROSENBLATT :* "Principles of Neurodynamics"
Spartan Books 1962.

[68] *G. SABAH, M. RAABE :*"Génération automatique d'un opérateur de prétraitement en traitement d'image"
Mémoire de l'Ecole Polytechnique - Janvier 1971.

[69] *A. L. SAMUEL :* "Some studies in Machine learning using the game of checkers ; II recent progress".
IBM Journal of R. and D. 11, p. 601-617, 1968.

[70] *H. P. SCHLAEPPI :* "A formal language for describing Machine Logic, Timing and Sequencing (LOTIS)"
IBM Research paper RZ-125. Décembre 24, 1963.

[71] *I. J. SCHÖNBERG* : "Contribution to the problem of approximation of equidistant data by analytic functions"
Quart. Journ. Appl. Math. t. 4, n°1. Janvier 1946.

[72] *C. E. SHANNON, W. WEAVER :* "The mathematical theory of communication"
University of Illinois Press, Urbana, Ill. 1949.

[73] *A. SHAW :* "A proposed Language for the Formal Description of Pictures"
CGS Memo 28, Stanford University - February 1967.

[74] *I. SHEINBERG :* "The INPUT 2 Document Reader (a new optical character recognition system)".
Pattern Recognition. Vol. 2, Nr 3, Sept. 1970.

[75] *R. N. SHEPARD :* "The analysis of proximities scaling with an unknown distance function"
I et II Psychometrika (1962).

[76] *H. A. SIMON :* "Experiments with a Heuristic Compiler"
J. ACM 1963.

[77] *J. C. SIMON :* "Les principes de la reconnaissance des formes"
Institut de Programmation - Novembre 1970.

[78] *J. C. SIMON :* "Prise et conservation de l'information, filtrages"
Publ. Int. - Institut de Programmation - Juin 1970.

[79] *J. C. SIMON, J. P. AUBINEAU :* "Function and signal representation in connection with precision"
Institut de Programmation - Paris, Août 1970.

[80] *J. C. SIMON, A. CHECROUN, C. ROCHE :* "Procédés de filtrage digital d'une image de lignes"
Compte rendu à l'Ac. des Sc. t. 272, p. 337-340
(25 Janvier 1971).

[81] *J. C. SIMON, A. CHECROUN :* "Procédés de filtrage numérique d'une image de lignes"
Automatisme - Mars 1972.

[82] *J. C. SIMON, A. CHECROUN, C. ROCHE :* "Comparaison des formes indépendante des déplacements, homothéties et petites déformations"
Compte rendu à l'Ac. des Sc. t. 270, p. 1607-1609
(15 Juin 1970).

[83] *J. C. SIMON, A. CHECROUN, C. ROCHE :* "A method of comparing two patterns independant of possible transformations and small distorsions"
IEEE Symposium on Syntactic Pattern Recognition - Chicago - Oct. 1970.
Publié dans Pattern Recognition. Vol. 4, Nr 1, Janvier 1972.

[84] *J. C. SIMON, C. ROCHE :* "Application of questionnaire theory to Pattern Recognition"
International Joint Conference on Artificial Intelligence - Londres - Septembre 1971.

[85] *J. C. SIMON, C. ROCHE, G. SABAH :* "On automatic generation of pattern recognition operators"
International Conference on Cybernetics - 6 Oct. 1972.

[86] *J. SLAGLE, R. LEE :* "Application of Game Tree Searching Techniques to Sequential Pattern Recognition"
Commun of the ACM - Février 1971.

[87] *P. H. SWAIN, K. S. FU :* "Nonparametric and linguistic Approaches to Pattern Recognition"
TR-EE 70-20. Purdue University, Lafayette, Indiana - Juin 1970.

[88] *M. TERRENOIRE :* "Un modèle mathématique de processus d'interrogation : les pseudoquestionnaires"
Thèse Sc. Math. Grenoble 1970.

[89] *Y. Z. TZYPKIN :* "Adaptation, learning and Self-organization in Automatic systems"
Automatika i Telemekhanika, Vol. 27, n°1, Janvier 1966.

[90] *S. WATANABE :* "A unified view of clustering algorithms"
IFIP Congress 1971, Ljubiana, Booklet TA-2.

[91] *T. WINOGRAD :* "Procedures as a Representation for Data in a Computer Program for Understanding Natural Language"
Ph. D. Thesis, Department of Mathematics, MIT, Août 1970.

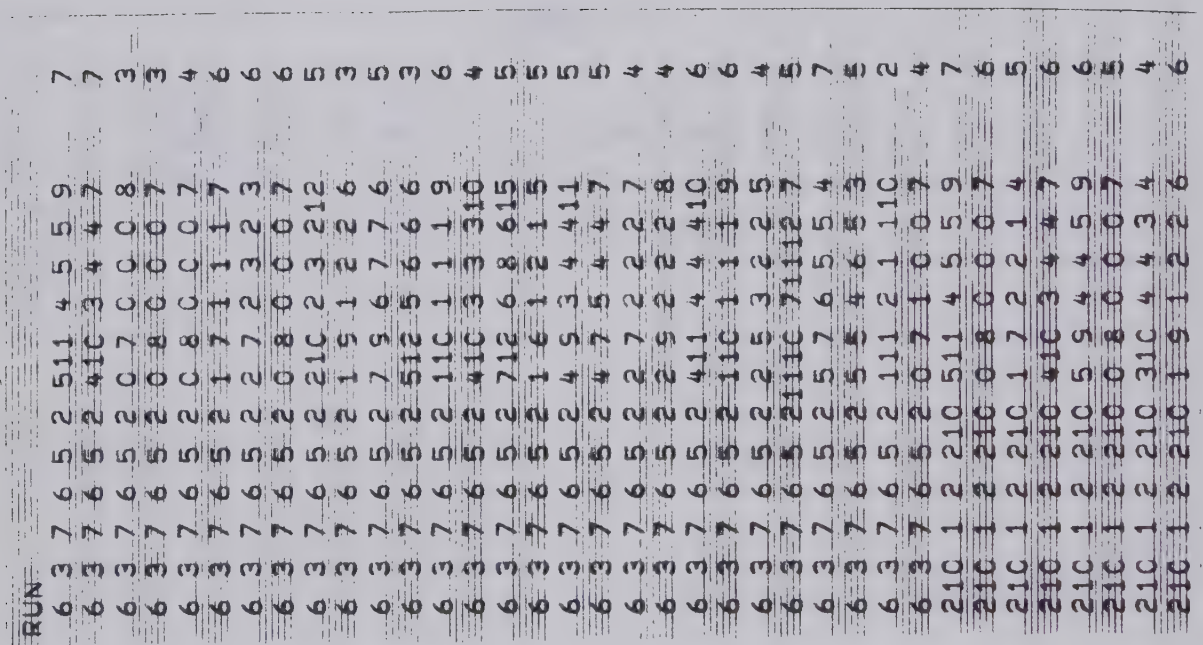
[92] *J. ARSAC, M. DUPUY, M. F. LE ROCH :* "Rapport sur l'emploi partagé d'un ordinateur"
(Institut de Programmation, convention DRME n° 6634589).

[93] *E. W. DIJSTRA :* "Cooperating sequential processes"
E. W. D. 123 - Heindhoven - Hollande 1965.

[94] *J. FELDMANN, J. GIPS , J. HORNING, S. REDER :* "Grammatical complexity and inference"
C. S. 125 - Memo AI 89 - Stanford University.

ANNEXE

SORTIE DU PROGRAMME DE GENERATION DE STRUCTURE,
APPLIQUE A L'APPRENTISSAGE DE LA NOTE DE RESSEMBLANCE
ENTRE CRÊTES (600 DONNEES)



Sous ensemble des données d'apprentissage : à chaque ligne 12 paramètres du couple de crêtes considéré, et la note donnée à ce couple par le professeur.

```
PROBABILITES DES FORMES: P(RFP(OM))
  1   2   3   4   5   6   7   8   9
 22  28  46  72  89 108  69  44  22
LES IMPORTANCES RELATIVES DES DIFFERENTES FORMES SONT:
  1   2   3   4   5   6   7   8   9
  1   1   1   1   1   1   1   1   1
NEGENTROPIE INITIALE : H(OM)=2.060

INFORMATION MUTUELLE DES OPERATEURS CANAUX
  LA DERNIERE COLONNE CORR. A I(O,OM)
    .000  .735 1.368 1.612 1.607  .905  .150  .112  .150  .149  .141  .110  .175
    .000  .000  .815  .782  .718 1.198  .122  .139  .115  .114  .119  .138  .115
    .000  .000  .000 1.316 1.244  .891  .124  .111  .118  .131  .128  .102  .142
    .000  .000  .000  .000 1.560  .974  .178  .135  .176  .167  .163  .131  .202
    .000  .000  .000  .000  .000  .888  .161  .132  .161  .156  .149  .120  .188
    .000  .000  .000  .000  .000  .000  .130  .150  .116  .136  .135  .143  .157
    .000  .000  .000  .000  .000  .000  .000  .661 1.449 1.665 1.838  .690  .196
    .000  .000  .000  .000  .000  .000  .000  .000  .748  .725  .694 1.089  .120
    .000  .000  .000  .000  .000  .000  .000  .000  .000 1.417 1.356  .701  .191
    .000  .000  .000  .000  .000  .000  .000  .000  .000  .000 1.641  .726  .197
    .000  .000  .000  .000  .000  .000  .000  .000  .000  .000  .000  .681  .192
    .000  .000  .000  .000  .000  .000  .000  .000  .000  .000  .000  .000  .135

APPROX. DE I(OI,OJ/OM)
    .000 -.446-1.051-1.235-1.245 -.572  .222  .182  .216  .223  .226  .200
    .000  .000 -.558 -.465 -.415 -.926  .189  .096  .190  .198  .188  .111
    .000  .000  .000 -.972 -.914 -.592  .215  .151  .214  .208  .205  .175
    .000  .000  .000  .000-1.170 -.615  .220  .186  .216  .232  .231  .205
    .000  .000  .000  .000  .000 -.542  .224  .176  .217  .230  .230  .202
    .000  .000  .000  .000  .000  .000  .224  .127  .232  .218  .214  .149
    .000  .000  .000  .000  .000  .000  .000 -.344-1.062-1.271-1.449 -.359
    .000  .000  .000  .000  .000  .000  .000  .000 -.438 -.412 -.383 -.834
    .000  .000  .000  .000  .000  .000  .000  .000  .000-1.029 -.974 -.375
    .000  .000  .000  .000  .000  .000  .000  .000  .000  .000-1.252 -.394
    .000  .000  .000  .000  .000  .000  .000  .000  .000  .000  .000 -.355

ON REGROUPE  7 ET 11 :INFO. APPR.=1.449
```

Début de l'apprentissage

choix des variables

servant d'entrée au premier opérateur créé

REGROUPEMENT DES ATTRIBUTS PLACES EN 7 ET 11
TABLEAU DES PROBABILITES MODIFIEES: P(REP(8 7),REP(811))
(REP(811)INDICE LES COLONNES,
REP(8 7)INDICE LES LIGNES)

114	C	C	0	C	C	C	C	0	C	C	C	C	C	C	C
C	67	11	C	C	C	C	C	0	C	C	C	C	C	C	C
2	1	63	6	C	C	C	C	0	C	C	C	C	C	C	C
C	C	C	39	3	C	C	C	0	C	C	C	C	C	C	C
C	C	C	6	57	C	0	C	0	C	C	C	C	C	C	C
C	C	C	0	C	35	8	C	0	C	C	C	C	C	C	C
C	C	C	C	C	4	8	5	0	C	C	C	C	C	C	C
C	C	C	C	C	C	12	20	5	C	C	C	C	C	C	C
C	C	C	C	C	C	C	C	0	10	C	C	C	C	C	C
C	C	C	C	C	C	C	C	0	11	C	C	C	C	C	C
C	C	C	0	C	C	C	C	0	C	C	C	C	C	C	C
C	C	C	C	C	C	C	0	0	C	C	C	4	C	C	C
C	C	C	C	C	C	C	C	0	C	C	C	C	C	C	C
C	0	C	0	C	C	0	C	0	C	C	C	0	9	C	C
C	C	C	C	C	C	C	C	0	C	C	C	C	C	C	C
C	0	C	C	C	C	C	C	0	C	C	C	C	C	C	C

I(7*11;8M)= .265

NOUVEL OPERATEUR N8 13 TRAVAILLANT SUR LES ATTRIBUTS PLACES EN 7 ET 11
TABLEAU DES REPONSES
(REP(811)INDICE LES COLONNES,
REP(8 7)INDICE LES LIGNES)

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	4	6	1	1	1	1	1	1	1	1	1	1	1	1	1
1	3	5	6	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	6	5	1	1	1	1	1	1	1	1	1	1	1
1	1	1	5	6	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	2	3	1	1	1	1	1	1	1	1	1
1	1	1	1	1	3	5	2	1	1	1	1	1	1	1	1
1	1	1	1	1	1	2	3	3	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	5	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	3	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	3	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

SA REPONSE SERA PLACEE EN 7
I(13;8M)= .181 A COMPARER A H(9M)=2.C6C

TABLEAU DES PROBABILITES:P(REP(013),REP(0M))

1	2	3	4	5	6	7	8	9	
15	9	8	15	8	19	8	19	15	116
5	9	5	10	12	12	6	2	0	61
C	1	9	14	20	6	2	1	0	53
1	4	5	12	8	9	14	10	4	67
C	2	5	15	10	28	23	4	3	90
1	3	14	6	31	34	16	8	0	113
C	0	0	0	0	C	0	0	0	0
C	0	0	0	0	C	0	0	0	0
C	0	0	0	0	C	0	0	0	0
C	0	0	0	0	C	0	0	0	0
C	0	C	0	0	C	0	0	0	0
C	0	0	0	0	C	0	0	0	0
C	0	0	0	C	C	0	C	0	0
C	0	0	0	0	C	0	0	0	0
C	0	C	0	0	C	0	0	0	0
C	0	0	0	0	C	0	0	0	0

APPROX. DE I(0I,0J;0M)
ET VALEUR DE I(0I,0M):DERN. COLONNE

.000	-.446	-1.051	-1.235	-1.245	-.572	.261	.182	.216	.223	.000	.200	.175
.000	.000	-.558	-.465	-.415	-.926	.217	.096	.190	.198	.000	.111	.115
.000	.000	.000	-.972	-.914	-.592	.250	.151	.214	.208	.000	.175	.142
.000	.000	.000	.000	-1.170	-.615	.277	.186	.216	.232	.000	.205	.202
.000	.000	.000	.000	.000	-.542	.272	.176	.217	.230	.000	.202	.188
.000	.000	.000	.000	.000	.000	.261	.127	.232	.218	.000	.149	.157
.000	.000	.000	.000	.000	.000	.000	-.129	-.681	-.824	.000	-.151	.181
.000	.000	.000	.000	.000	.000	.000	.000	-.438	-.412	.000	-.834	.120
.000	.000	.000	.000	.000	.000	.000	.000	.000	-1.029	.000	-.375	.191
.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	-.394	.197
.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.192
.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.135

ON REGROUPE 1 ET 5 :INFO. APPR.=1.245

Choix des variables d'entrée du

2ème opérateur...

REGROUPEMENT DES ATTRIBUTS PLACES EN 1 ET 7
TABLEAU DES PROBABILITES MODIFIEES: P(REP(B 1),REP(B 7))
(REP(B 7)INDICE LES COLONNES,
REP(B 1)INDICE LES LIGNES)

8	4	6	32	21	14	0	0	0	0	0	0	0	0	0	0
5	3	1	20	5	3	0	0	0	0	0	0	0	0	0	0
13	10	7	19	26	23	0	0	0	0	0	0	0	0	0	0
21	11	14	30	44	25	0	0	0	0	0	0	0	0	0	0
6	5	9	38	12	5	0	0	0	0	0	0	0	0	0	0
4	6	2	18	16	14	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I(1* 7;BM)= .652

NOUVEL OPERATEUR NB 23 TRAVAILLANT SUR LES ATTRIBUTS PLACES EN 1 ET 7
TABLEAU DES REPONSES
(REP(B 7)INDICE LES COLONNES,
REP(B 1)INDICE LES LIGNES)

1	2	3	4	3	6	1	1	1	1	1	1	1	1	1	1
1	7	5	4	4	3	1	1	1	1	1	1	1	1	1	1
1	2	9	7	8	8	1	1	1	1	1	1	1	1	1	1
6	2	3	3	7	6	1	1	1	1	1	1	1	1	1	1
1	6	5	4	1	6	1	1	1	1	1	1	1	1	1	1
9	3	2	9	3	6	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

SA REPONSE SERA PLACEE EN 1
I(23;BM)= .497 A COMPARER A H(BM)=2.C6C

...Fin de l'apprentissage
dernier opérateur créé

```
TABLEAL DES PROBABILITES:P(REP(A23),REP(9M))
 1   2   3   4   5   6   7   8   9
 9   2   1   6   7   2   2   C   0  29
 4   6   C   0   C   C   C   0   0  10
 C   3  15  25  33   6   2   C   0  84
 C   C   5   C   8  13   0   1   0  27
 1   9  10  18  13  34  10   1   0  96
 0   C   3  10   7  32  12   2   0  66
 C   C   C   6  14  1C  18   1   0  49
 C   0  11   4   5   C   7  17   0  44
 8   8   1   3   2  11  18  22  22  95
SUR UN TOTAL DE  500 ECHANTILLONS


TEST DE L OPERATEUR CREE: SUR LES DONNEES D APPRENTISSAGE

 6 3 7 6 5 2 511 4 5 5 9      7    6
 6 3 7 6 5 2 41C 3 4 4 7      7    6
 6 3 7 6 5 2 0 7 C 0 C 8      3    5
 6 3 7 6 5 2 0 8 0 0 0 7      3    5
 6 3 7 6 5 2 0 8 C C C 7      4    5
 6 3 7 6 5 2 1 7 1 1 1 7      6    5
 6 3 7 6 5 2 2 7 2 3 2 3      6    6
 6 3 7 6 5 2 0 8 0 0 0 7      6    5
 6 3 7 6 5 2 21C 2 3 212      5    3
 6 3 7 6 5 2 1 9 1 2 2 6      3    4
 6 3 7 6 5 2 7 9 6 7 7 6      5    3
 6 3 7 6 5 2 512 5 6 6 6      3    3
 6 3 7 6 5 2 11C 1 1 1 9      6    5
 6 3 7 6 5 2 41C 3 3 310      4    6
 6 3 7 6 5 2 712 6 8 615      5    5
 6 3 7 6 5 2 1 6 1 2 1 5      5    3
 6 3 7 6 5 2 4 9 3 4 411      5    4
 6 3 7 6 5 2 4 7 5 4 4 7      5    3
 6 3 7 6 5 2 2 7 2 2 2 7      4    6
 6 3 7 6 5 2 2 9 2 2 2 8      4    6
 6 3 7 6 5 2 411 4 4 41C      6    6
 6 3 7 6 5 2 11C 1 1 1 9      6    5
 6 3 7 6 5 2 2 5 3 2 2 5      4    5
```

Table de confusion de la structure engendrée sur l'ensemble d'apprentissage. Comparaison sur chaque donnée d'apprentissage de la note donnée par le professeur (13ème colonne) et de la note donnée par la structure (14ème colonne)

Suite de la comparaison

TEST DE L OPERATEUR CREE: SUR LES DONNEES SUIVANTES

6 3 7 6 5 2 5 11 4 5 5 9	7	6
6 3 7 6 5 2 41C 3 4 4 7	7	6
6 3 7 6 5 2 C 7 C C C 8	3	5
6 3 7 6 5 2 C 8 C C C 7	3	5
6 3 7 6 5 2 C 8 C C C 7	4	5
6 3 7 6 5 2 1 7 1 1 1 7	6	5
6 3 7 6 5 2 2 7 2 3 2 3	6	6
6 3 7 6 5 2 C 8 C C C 7	6	5
6 3 7 6 5 2 21C 2 3 212	5	3
6 3 7 6 5 2 1 9 1 2 2 6	3	4
6 3 7 6 5 2 7 9 6 7 7 6	5	3
6 3 7 6 5 2 512 5 6 6 6	3	3
6 3 7 6 5 2 11C 1 1 1 9	6	5
6 3 7 6 5 2 41C 3 3 310	4	6
6 3 7 6 5 2 712 6 8 615	5	5
6 3 7 6 5 2 1 6 1 2 1 5	5	3
6 3 7 6 5 2 4 9 3 4 411	5	4
6 3 7 6 5 2 4 7 5 4 4 7	5	3
6 3 7 6 5 2 2 7 2 2 2 7	4	6
6 3 7 6 5 2 2 9 2 2 2 8	4	6
6 3 7 6 5 2 411 4 4 410	6	6
6 3 7 6 5 2 11C 1 1 1 9	6	5
6 3 7 6 5 2 2 5 3 2 2 5	4	5
6 3 7 6 5 2111C 7111 2 7	5	3
6 3 7 6 5 2 5 7 6 5 5 4	7	5
6 3 7 6 5 2 5 5 4 6 5 3	5	5
6 3 7 6 5 2 111 2 1 110	2	3
6 3 7 6 5 2 C 7 1 C 6 7	4	3
21C 1 2 210 511 4 5 5 9	7	6
21C 1 2 210 C 8 C C C 7	6	5
21C 1 2 21C 1 7 2 2 1 4	5	3
21C 1 2 21C 41C 3 4 4 7	6	6
21C 1 2 21C 5 9 4 4 5 9	6	3
21C 1 2 21C C 8 C C C 7	5	5

Comparaison sur les données de test

```
PROBABILITES DE TEST

   1  2  3  4  5  6  7  8  9
C  0  0  1  0  0  0  0  0  0
C  0  C  0  0  C  0  0  0  0
0  1  2  8 10  1  0  0  0  0
C  0  1  0  3  7  0  0  0  0
C  0  2  3  3  5 11  2  0  0  0
C  0  C  0  6  4 14  7  0  0  0
C  0  0  0  0  2  2  3  0  0  0
C  0  0  C  1  1  C  0  1  0  0
C  0  0  0  0  0  0  0  0  0  0

SUR UN TOTAL DE 1C1 ECHANTILLONS
*STOP* 0

TOTAL JOB TIME                    9.333
  PROCESSOR EXECUTION TIME        1.044
PROCESSOR I/O TIME                1.352
  PROCESSOR OVERHEAD TIME          .628
USER EXECUTION TIME               3.528
  USER I/O TIME                   2.133
USER OVERHEAD TIME                 .648
# OF CARDS READ                   1242
# OF CARDS PUNCHED                   2
# OF PROCESSOR PAGES OUT            33
# OF USER PAGES OUT                 80
# OF DIAGNOSTIC PAGES OUT            C
# OF SCRATCH TAPES USED              0
# OF SAVE TAPES USED                 0
# OF DISK READS AND WRITES         431
# OF DISC READS AND WRITES        6396
TEMPORARY DISC SPACE USED             C
PERMANENT DISC SPACE USED            58
ACCUM. PERM. DISC SPACE USED          C
```

Table de confusion sur les données de test

Temps d'exécution du programme (C2I 10070)

ACADÉMIE DE PARIS

UNIVERSITÉ DE PARIS VI

Place Jussieu - 75230 PARIS-CEDEX 05

SERVICE DE LA SCOLARITÉ

Bureau des Enseignements du 3e cycle

TÉLÉPHONE 336-25-25 - 325-12-21

PERMIS D'IMPRIMER

DOCTORAT : DOCTORAT ES SCIENCES

THESE DE : SCIENCES MATHEMATIQUES

NOM : Monsieur ROCHE

Prénoms : Claude

né le 18 décembre 19.44 à LILLE Dépt : NORD

1ère THESE : "Information utile en reconnaissance des formes et en compression des données. Application à la génération automatique de systèmes de reconnaissance optique et acoustique."

2e THESE (Propositions faites par l'Université) :

PRESIDENT : M. LELONG

EXAMINATEURS : M. MALAVARD
M. SIMON
M. PICARD
M. MERCOUROFF

Vu et Approuvé

PARIS, le 4 janvier 1973

Le Président

de l'Université de PARIS VI